

Tạp chí
VN **I**

Xuân Ất Tỵ



20
25

LỜI NÓI ĐẦU

Quý độc giả thân mến,

Tạp chí VNOI - Xuân Ất Ty 2025 chính thức ra mắt, tiếp nối hành trình chia sẻ tri thức, kinh nghiệm và những câu chuyện từ cộng đồng Tin học Việt Nam. Đây là năm thứ ba Tạp chí được xuất bản, mang theo sứ mệnh lan tỏa những giá trị cốt lõi và cảm hứng từ thế giới Tin học và Lập trình, ban biên tập với mong muốn rằng có thể kết nối và lan tỏa đam mê Tin học mãnh liệt đến từng độc giả.

Lấy cảm hứng từ sự thành công của hai số trước, Tạp chí VNOI 2025 tiếp tục kế thừa và phát triển, mang đến những nội dung sâu sắc, đa dạng và dễ tiếp cận hơn với nhiều đối tượng khán giả. Trong số đặc biệt này, Ban biên tập tự hào giới thiệu 2 bài viết học thuật thú vị nhưng không quá khô khan, được tuyển chọn kỹ lưỡng từ nhiều bài viết gửi về trên khắp mọi miền đất nước. Đồng thời, tạp chí còn có sự góp mặt của 3 vị khách mời đặc biệt qua những bài phỏng vấn độc quyền, mỗi khách mời đều mang nhiều dấu ấn cá nhân xuyên suốt dòng chảy lịch sử Tin học Việt Nam và mỗi người đều đem đến cho độc giả những góc nhìn đa chiều về thế giới Tin học qua những lời chia sẻ và trải nghiệm của bản thân.

Nhân dịp Xuân Ất Ty 2025, Ban biên tập xin gửi lời tri ân chân thành đến các tác giả, khách mời, và quý độc giả đã luôn đồng hành cùng Tạp chí VNOI trong suốt thời gian qua. Sự ủng hộ và đóng góp của các bạn chính là nguồn động lực to lớn để Tạp chí ngày càng hoàn thiện và phát triển.

Chào đón mùa xuân mới, Ban biên tập kính chúc quý độc giả một năm tràn đầy sức khỏe, hạnh phúc và thành công rực rỡ!

BAN BIÊN TẬP



MỤC LỤC

VNOI – Nhìn lại 2024	3
VNOI CUP 2024 - Cuộc đua trí tuệ không giới hạn	11
OLP'24 & ICPC Asia Hanoi 2024 - Nhiều dấu ấn khó quên	15
Thuật toán tìm số nguyên tố của Dijkstra	18
Tổng quan về cấu trúc dữ liệu	23
Phỏng vấn Nguyễn Khánh Phúc	30
Phỏng vấn Nguyễn Tấn Sỹ Nguyên	40
Phỏng vấn bác Nguyễn Long	53



VNOI – Nhìn lại 2024

Được thành lập với mục đích thúc đẩy phong trào Tin học, Câu lạc bộ Olympic Tin học Việt Nam - VNOI là nơi giao lưu, chia sẻ dành cho học sinh, sinh viên đam mê lập trình thi đấu nói riêng và lĩnh vực công nghệ thông tin nói chung. Trải qua hơn 16 năm hoạt động, VNOI đã có nhiều đóng góp nổi bật cho nền tin học nước nhà. Đặc biệt, trong năm 2024 vừa qua, VNOI đã có nhiều hoạt động mang dấu ấn khó quên. Hãy cùng chúng mình điểm lại những hoạt động và thành tựu của VNOI trong năm vừa qua thông qua bài viết này nhé.

Đồng hành với ICPC¹ và Olympic Tin học Sinh viên 2024

Năm 2024 đánh dấu năm thứ tư VNOI đồng hành cùng các kỳ thi ICPC và Olympic Tin học Sinh viên. Kế thừa và phát triển công nghệ, kỹ thuật và kinh nghiệm từ các năm trước, VNOI tiếp tục hỗ trợ tổ chức các sự kiện diễn ra thuận lợi, tốt đẹp.

Vào tháng 3, ICPC Asia Pacific Championship lần đầu tiên được tổ chức, do Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội đăng cai, đã khép lại với thành công rực rỡ. Đây là kỳ thi mang ý nghĩa đặc biệt quan trọng, quyết định tư cách tham dự vòng Chung kết thế giới ICPC World Final của các đội tuyển. Giải đấu quy tụ 195 sinh viên của 65 đội tuyển và 50 huấn luyện viên, đại diện cho 39 trường đại học hàng đầu từ Nhật Bản, Hàn Quốc, Đài Loan, Singapore, Thái Lan, Indonesia và Việt Nam.



BTC và tập thể thí sinh, các huấn luyện viên chụp ảnh lưu niệm.

Đáng chú ý, đây cũng là lần đầu tiên VNOI trực tiếp hỗ trợ kỹ thuật cho một sự kiện tầm cỡ khu vực và quốc tế, với hệ thống giám sát thi do VNOI phát triển được hội đồng chỉ định sử dụng, nhằm đảm bảo tính công bằng và minh bạch cho kỳ thi.

Tháng 12, ICPC Asia Hanoi 2024 đã diễn ra thành công với nhiều điểm sáng trong khâu tổ chức và xuyên suốt kỳ thi. Đặc biệt có thể kể đến việc hỗ trợ in mã nguồn trong quá trình thi, chỉ mất

một thời gian rất ngắn để bản in tới được tay thí sinh từ khi yêu cầu được gửi đi.



VNOI tại ICPC Asia Hanoi 2024

VNOI Roadmap

Lập trình thi đấu là môn học xuất hiện muộn hơn các môn khoa học khác và là môn học đặc thù. Không như các môn văn hóa phổ thông, môn học này ban đầu được giảng dạy trong chương trình chuyên Tin, sau đó được điều chỉnh, giảm tải để phù hợp với chương trình phổ thông. Vì vậy, các tài liệu tham khảo dành cho người tự học hiện nay không nhiều và chưa được hệ thống hóa, đồng thời số thầy cô có thể giảng dạy bộ môn này cũng còn hạn chế. Do đó, các bạn học sinh, sinh viên muốn tự học, tự phát triển để tham gia các kỳ thi gặp rất nhiều khó khăn, đặc biệt trong việc xác định lộ trình học tập.

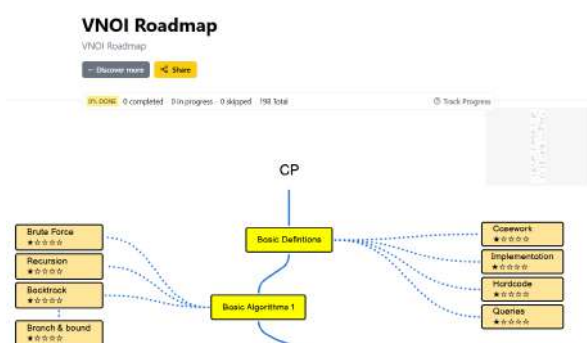
Thấu cảm khó khăn mà các bạn gặp phải, VNOI đã thực hiện dự án VNOI Roadmap, giúp các bạn tự định hướng trong việc học tập, chuẩn bị kiến thức cho các kỳ thi quan trọng. Roadmap là một lộ trình hoàn chỉnh bao gồm các chủ đề được đánh giá độ khó từ 1 đến 5 sao, dành cho mọi đối tượng học sinh, sinh viên hay những người đam mê lập trình thi đấu.

¹ICPC: International Collegiate Programming Contest - Cuộc thi Lập trình Quốc tế lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu.



VNOI Roadmap

VNOI Roadmap là sự đúc kết kinh nghiệm của các thế hệ đi trước, đã trải và vượt qua nhiều cuộc thi trong suốt những năm vừa qua. VNOI hy vọng Roadmap 2024 sẽ giúp việc tự học Lập trình thi đấu của các bạn trở nên dễ dàng hơn, đóng góp thật nhiều vào phong trào Tin học nước nhà.

VNOI Roadmap 2024 tại roadmap.sh

VNOI Roadmap đã được tải lên và công khai tại roadmap.sh², một trang web nguồn mở cung cấp lộ trình chi tiết về việc tự học các kỹ năng khác nhau trong lĩnh vực Công nghệ thông tin.

Phát triển dịch vụ VNOJ

Năm 2024 ghi dấu sự tăng trưởng đáng kể về lưu lượng sử dụng VNOJ, với số lượt nộp trung bình khoảng 9700 lượt/ngày, trong đó 68% đến từ các Tổ chức. Đây vừa là cơ hội vừa là thách thức cho đội ngũ VNOI: cơ hội vì hệ thống được tin dùng rộng rãi bởi các bạn học sinh, sinh viên và các thầy, cô trên cả nước; thách thức vì cần bảo đảm hiệu suất, tốc độ ổn định trong bối cảnh lượng truy cập không ngừng gia tăng.

²<https://roadmap.sh/r/vnoi-roadmap>

³<https://www.facebook.com/share/p/1BRjGDxqDC/>

⁴https://drive.google.com/file/d/1bVHDJOer2yzSILHgaO-Z-D9_trQxdjpV/view



Quy định sử dụng tính năng tổ chức

Để duy trì hệ thống lâu dài và bền vững, VNOI đã tăng công suất máy chủ lên gấp rưỡi, đồng thời ban hành bộ quy định cho việc sử dụng tính năng Tổ chức trên VNOJ. Hướng dẫn chi tiết và danh sách những câu hỏi thường gặp đã được VNOI đăng tải, giải đáp trên [fanpage chính thức của VNOI](#)³.

Song song đó, nhằm khắc phục các hạn chế của tính năng Tổ chức và tăng cường hiệu quả dạy – học, VNOI quyết định cung cấp dịch vụ hỗ trợ duy trì, vận hành một hệ thống chấm bài độc lập cho các tổ chức giáo dục có nhu cầu. Thông tin đầy đủ về dịch vụ này có trong [thông báo hình thức về việc cung cấp trang chấm bài độc lập](#)⁴.

VNOI Cup 2024

Năm 2024, VNOI tiếp tục tổ chức VNOI Cup, một kỳ thi thường niên thu hút số lượng đông đảo các thí sinh có profile “khủng” tham gia tranh tài. VNOI Cup năm nay có nhiều điểm mới so với hai lần tổ chức trước: Không còn giới hạn độ tuổi tham dự và số lượng thí sinh được tham dự vòng chung kết tăng lên 24, gấp đôi so với năm 2023.



Lộ trình VNOI Cup 2024

Các thí sinh VNOI Cup 2024 phải trải qua hai vòng loại Online diễn ra trên nền tảng VNOJ, từ đó

lấy cơ sở để lựa chọn những thí sinh xuất sắc nhất tham dự vòng Chung kết diễn ra tại Trường Đại học Khoa học tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh.

Sau hai vòng thi, danh sách thí sinh tham dự vòng Chung kết VNOI Cup 2024 đã lộ diện, trong đó có rất nhiều gương mặt quen thuộc từ hai mùa giải trước và cũng xuất hiện rất nhiều “nhân tố” mới. Đặc biệt, năm 2024 là lần đầu tiên Ban tổ chức ghi nhận có thí sinh cấp Trung học Cơ Sở giành được suất tham dự vòng Chung kết, đó là bạn Nguyễn Khánh Phúc, khi diễn ra Chung kết, bạn đang là học sinh lớp 8 của trường THCS Trương Hán Siêu, Ninh Bình.



DANH SÁCH THÍ SINH THAM DỰ			
CHUNG KẾT VNOI [CUP] 2024			
duongnc000	noobcoder	chemthan	khanhphucscratch
fextivity	6aren	flashmt	toib
gawr_gura	hoiwo_po1w	tuanninh17032006	tungvh04
marvinthang	Yurushia	kilkuku	Panda
minhcool	LeKienThanh69	hoangxuanbach	DeMen100ns
socp1te	neko_nyaaaaaaaaa	cuom1999	powder

Danh sách thí sinh tham dự Chung kết VNOI Cup 2024

Trong 4 tiếng đầu của vòng Chung kết, Ban tổ chức phát trực tiếp cùng với bình luận thi trên fanpage để tất cả mọi người có thể theo dõi diễn biến kỳ thi từ xa. Riêng giờ thi cuối, việc phát trực tiếp được tạm dừng để đảm bảo bảo mật kết quả kỳ thi.



Chung kết VNOI Cup 2024 tại Trường Đại học Khoa học tự nhiên, ĐHQG Thành phố Hồ Chí Minh

Sau 5 tiếng diễn ra kỳ thi, top 3 thí sinh xuất sắc nhất của VNOI Cup 2024 đã chính thức lộ diện. Tất cả đều là những gương mặt quen thuộc với cộng đồng Tin học Việt Nam. Vượt qua nhiều đối thủ mạnh, Trần Xuân Bách xuất sắc giành ngôi Quán quân; huy chương Bạc thuộc về Lê Phước

Định; và giành được huy chương Đồng là Nguyễn Tấn Sỹ Nguyên. Các thí khác còn lại cũng đã rất xuất sắc khi đã vượt qua rất nhiều thí sinh để tiến vào vòng Chung kết, và đều nhận được những phần quà đặc biệt là những chiếc áo VNOI có in tên riêng của các bạn.

VNOI và các dự án cộng đồng 2024

Bedao Contest

Bedao Contest là một dự án được thành lập vào đầu năm 2020 nhằm mục đích tổ chức các kỳ thi, tạo ra sân chơi cho những bạn trẻ yêu thuật toán và coding. Bedao và VNOI chính thức bắt tay vào tháng 8/2021 và sát nhập như hiện nay.



Bedao OI Contest 5 đón nhận sự tham gia của hơn 500 bạn thí sinh

Trong năm 2024, VNOI tiếp tục dự án Bedao Contest. Đặc biệt, đội ngũ Bedao Contest tiếp tục tổ chức 3 kỳ Bedao OI trước thêm kỳ thi Học sinh giỏi Quốc gia, nhằm giúp các bạn học sinh có một cơ hội luyện tập với trải nghiệm tương tự kỳ thi chính thức.

Đồng hành với các bạn học sinh trong kỳ thi Học sinh giỏi Quốc gia

Trong ba năm trở lại đây, trước thêm kỳ thi Học sinh giỏi Quốc gia, VNOI đều duy trì việc liên hệ các anh, chị tiền bối để xin những ý kiến chia sẻ về việc thi cử và đăng tải lên fanpage cho các bạn học sinh tham khảo.

⁵HSGQG: là viết tắt của kỳ thi Học sinh giỏi Quốc gia.



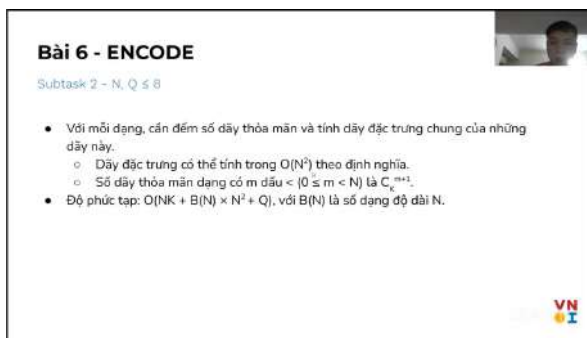
Chuyên mục kinh nghiệm và lời khuyên cho kỳ thi HSGQG⁵ hàng năm



Đợt tuyển TNV lớn nhất trong năm của VNOI

Sau hai ngày thi, VNOI tổ chức buổi chữa đề thi HSGQG môn Tin học, phát trực tiếp trên fanpage. Lời giải cho các bài tập được trình bày trong buổi chữa đề được chuẩn bị kỹ lưỡng và kịp thời, do đó nhận được sự hưởng ứng của cộng đồng, cao điểm buổi chữa có thể lên tới gần 600 lượt xem.

Nhằm gắn kết tình cảm giữa Tình nguyện viên với nhau, các hoạt động sinh hoạt được tổ chức thường xuyên vào các dịp đặc biệt tại hai thành phố lớn là Hà Nội và Thành phố Hồ Chí Minh.



Phần chữa bài 6 đề thi HSGQG 2025



Sinh hoạt tình nguyện viên tại Hà Nội

VNOI Wiki

Trong năm qua VNOI vẫn duy trì dự án VNOI Wiki với hơn 19 bài viết chất lượng về các chủ đề trong Tin học đã được xuất bản bởi đội ngũ VNOI Wiki, nhiều kỹ thuật, thuật toán được đưa tới cộng đồng. Nổi bật nhất là việc nâng cấp giao diện VNOI Wiki giúp truyền tải hiệu quả hơn nội dung các bài viết cho cộng đồng.



Sinh hoạt tình nguyện viên tại Thành phố Hồ Chí Minh

Tình nguyện viên Gen 4

Để duy trì các dự án phục vụ cộng đồng, VNOI rất cần sự giúp đỡ của các bạn Tình nguyện viên. Trong năm 2024, VNOI đã kết thúc và cấp giấy chứng nhận cho các bạn Tình nguyện viên thế hệ thứ 3 và vận động đăng ký Tình nguyện viên thế hệ thứ 4.

Ngoài gắn kết tình cảm tại các buổi sinh hoạt giữa các team TNV, VNOI còn tạo điều kiện cho các xây dựng tình yêu quê hương, đất nước qua các buổi tham quan bảo tàng hay các di tích lịch sử độc đáo tại thành phố Hà Nội & Hồ Chí Minh.



TNV tại Hà Nội tham quan di tích lịch sử Nhà tù Hỏa Lò



Giao lưu giữa các TNV tại TP.HCM



TNV tại TP.HCM tham quan Bảo tàng lịch sử Việt Nam

Những hoạt động của VNOI đều có dấu ấn đậm nét đến từ các bạn tình nguyện viên đã và đang âm thầm đóng góp chung cho nền Tin học nước nhà. Những dự án của VNOI sẽ không thành công nếu không có sự giúp sức và ủng hộ đến từ các bạn. VNOI xin chân thành cảm ơn các bạn, hy vọng rằng tinh thần của các bạn sẽ được lan tỏa tới những thế hệ tiếp theo.

Chia sẻ của các bạn TNV Gen 4

Trần Quang Trường - Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh

"Chào mọi người, em là Trần Quang Trường. Hiện tại, em đang là sinh viên năm nhất của trường Đại học Công nghệ Thông tin, ĐHQG-HCM và đã theo chân cùng với VNOI được 2 năm làm tình nguyện viên. Bắt đầu vào lúc em viết đơn ứng tuyển làm TNV VNOI, vào khoảng tháng 8 năm 2023. Em lo lắng khi điền vào đơn ứng tuyển. Vì yêu cầu ứng tuyển, không chỉ cần một background tốt mà cần có niềm đam mê, sự chăm chỉ và thái độ làm việc tốt, bởi thế việc được chọn để nằm trong 30 bạn xuất sắc nhất trong nhiều lá đơn khác nhau là rất đối khó khăn.

Với sự cố gắng và sự chờ đợi của mình, em đã không thất vọng khi sau khoảng một tháng, em đã nhận được mail trúng tuyển. Ngày em biết em đậu, em vui như được mùa, vui như cách em đậu vào trường Chuyên cấp 3 vậy!! Bởi việc đậu trong hơn hàng trăm đơn ứng tuyển của những bạn có cùng niềm đam mê khác là thử thách khó khăn.

Sau khi vào VNOI, em cùng các bạn TNV khác trong buổi meeting đầu khoá được hướng dẫn các hoạt động chung của nhóm và công việc chuyên môn của team em tham gia. Ví dụ như team contest chuyên về tổ chức tạo các kì thi để học tập cũng như luyện tập, team wiki chịu trách nhiệm việc soạn các bài viết về các thuật toán để đưa tin học nước nhà gần hơn với các bạn học sinh sinh viên, v.v.

Sau buổi meeting, em có khoảng thời gian để làm quen với các công việc chuyên môn. Ở khoảng thời gian đầu này, em gặp đôi chút khó khăn khi chưa nắm rõ được cách sử dụng của Polygon lúc ở team contest hoặc không hiểu được các cú pháp của Markdown khi làm ở team wiki. Nhưng may mắn thay, em được anh chị trong team đã hỗ trợ rất nhiều để giúp em vượt qua giai đoạn làm quen đó. Nhờ quy trình làm việc theo nhóm nhỏ, em đã làm quen được một số bạn TNV khác. Các bạn này rất giỏi ở lĩnh vực lập trình thi đấu nói riêng và ngành công nghệ thông tin nói chung, và nhờ đó mà em đã học hỏi được rất nhiều thứ từ các bạn. Fun fact, teammate ICPC cùng với em sắp tới đi thi ICPC Asia Pacific Championship ở Singapore đều là TNV team contest chung với em.

Cứ mỗi tháng, VNOI tổ chức buổi meeting tổng kết tháng vừa qua. Điều em ấn tượng nhất ở buổi meeting này chính là bản nhạc quay xổ số



huyền thoại khi quay số cho ai may mắn được nhận thưởng cho tháng vừa qua. Càng làm nhiều sẽ tăng tỉ lệ trúng khi tăng số tên xuất hiện trên bảng xổ số, nhưng trúng hay không sẽ tùy thuộc vào may mắn của bạn. Và rất "may mắn" vì hầu như tháng mình đều nhận được thưởng.



Sau khi quay xổ số xong, VNOI thường tổ chức chơi game chung hoặc văn nghệ các kiểu. Và bài văn nghệ em ấn tượng nhất lại buồn cười chính là bài em biểu diễn vào Tết năm ngoái: [QioCas mặc áo VNOI lắc lư trong điệu nhạc Tết](#)⁶. Tuy em múa không đẹp nhưng tiết mục múa của em ngày hôm đó đã để lại những ấn tượng khó quên đối với mọi người vào dịp Tết năm ấy.

Ngoài những buổi meeting cuối tháng, VNOI thường mời những anh lớn có 'máu mặt' trong ngành để chia sẻ và bàn về những đề tài như hướng nghiệp đi làm, làm sao để có một buổi phỏng vấn thật tốt, cách viết CV, hoặc chia sẻ những kinh nghiệm thi cử như kì thi Quốc gia, ICPC. Qua các buổi như vậy, em đã học hỏi những kiến thức giúp em phát triển trên con đường của bản thân mình.

Vào dịp Tết nguyên đán, VNOI cũng gửi quà chúc mừng tết là những bao lì xì xinh xắn cùng với chiếc áo VNOI. Về chiếc áo, đây như một ghi nhận hành trình của em tại VNOI. Mặc lên mà em cứ ngỡ mặt long bào, ngạo nghễ khi đi khoe với đám bạn cũng như các bà hàng xóm. 'Áo này con được cộng đồng Olympic tin học Việt Nam tặng á các cô chú.'! Về bao lì xì cũng có một câu chuyện vui, sau khi lên trường sau Tết em có tổ chức contest để mọi người trong đội tuyển tham gia sau đó tặng những phong bì VNOI với mệnh giá tùy thuộc vào số bài đã AC. Oách xà lách vô cùng.

Một kì TNV khoảng 3-4 lần, VNOI tổ chức buổi gặp mặt trực tiếp tại hai đầu Nam Bắc. Thường sẽ là buổi gặp mặt ăn uống, đi chơi đâu đó, hoặc như năm nay em được đi thăm quan các di tích lịch

sử. Lúc đầu gặp mặt em có hơi ngại vì hướng nội nhưng nhờ sự nhiệt tình giúp đỡ của anh chị nên em đã đỡ hơn. Qua buổi gặp mặt trực tiếp, em đã làm quen thêm được những người bạn ở team khác mà bình thường em chưa tiếp xúc. Tuy là một buổi ngắn ngủi nhưng đã đọng lại trong em nhiều điều quý giá.



Những món quà đặc biệt đến từ VNOI dành cho những bạn tình nguyện viên xuất sắc

Nhiều người thường hay bảo: "Làm TNV có gì đâu, làm chi cho cực". Đúng, đã làm một TNV là trên tinh thần tự nguyện không màng lợi ích. Tuy nhiên không vì thế mà bảo làm TNV không được gì, nhờ làm TNV VNOI em đã có cơ hội được làm quen được những người bạn mới, được học hỏi được nhiều kiến thức qua các buổi seminar, và cũng như rèn giũa những kĩ năng cho bản thân (kĩ năng làm việc nhóm, kĩ năng giao tiếp, kĩ năng múa nhảy, ...). Vậy còn "Làm TNV có cực không?" Đó là câu hỏi của bản thân em trước khi điền vào đơn ứng tuyển. Ở hai giai đoạn em viết đơn ứng tuyển là năm lớp 12 chuẩn bị cho kì thi HSGQG và đầu năm nhất chưa bắt nhịp được với cuộc sống sinh viên, thời điểm khó khăn trong việc ôn thi và chật vật với cuộc sống còn đầy điều bỡ ngỡ, em đã khá lo lắng khi sợ việc làm TNV ảnh hưởng tới bản thân. Nhưng nỗi lo ấy đã biến mất sau khi thật sự làm TNV, bởi mỗi tháng em được giao khá ít việc nên nhìn chung việc làm không quá áp lực. Việc được làm bởi sự đam mê dành tin học cũng là một

⁶<https://www.youtube.com/watch?v=lmGf6zX7e2o>



phần khiến em thấy làm TNV vui hơn hết. Ngoài ra, VNOI rất quan tâm tới các TNV do đó em không những không thấy thiệt thòi mà còn được giúp đỡ nhiều trên quá trình phát triển của mình.

Và cứ như vậy, sau gần 10 tháng làm TNV, vào tháng 5 năm 2023, VNOI tổ chức cuối khoá gen 3 và gửi lời cảm ơn tới em cùng những bạn TNV cùng khoá vì đã đồng hành với VNOI trong suốt thời gian qua. Với món quà cuối cùng là những bộ sticker, hình dán, móc khoá và không thể thiếu chính là giấy chứng nhận TNV. Một tờ giấy minh chứng cho quá trình đồng hành với VNOI trong một năm vừa qua. Để mà nói, hành trình đồng hành với VNOI của em là một chặng đường quan trọng trên con đường phát triển bản thân mình. Nhờ VNOI, em học được nhiều thứ, rèn luyện, trau dồi những kỹ năng và tích lũy cho bản thân những hành trang cần thiết cho sau này.

Một lần nữa, em xin cảm ơn VNOI vì tất cả. Em chúc VNOI tiếp tục phát triển để đưa nền tin học Việt Nam gần hơn với học sinh. Để mỗi nhà 5 mét vuông 1 Tourist, để Việt Nam đạt được những thành tích cao hơn trong những cuộc thi quốc tế. Một cái Tết nữa lại đến, em xin chúc mọi người trong VNOI cũng như tất cả bạn đang học tin học trên cả nước một cái Tết ấm no, tràn đầy niềm vui, gặp nhiều may mắn và thành công trên con đường sự nghiệp ạ."

Châu Tấn Phát - 12CTin, Trường THPT chuyên Trần Đại Nghĩa, TP.HCM

"Lúc mới bước vào cấp 3, vốn kiến thức về tin học của mình khá hạn hẹp, đặc biệt về CP⁷, nên trong khoảng thời gian đầu cấp này mình quyết định tìm tài liệu để học những kiến thức cần có cho CP nói chung cũng như cho bộ môn Tin chuyên nói riêng. VNOI lúc ấy như người thầy của mình, thấp sáng trong mình một niềm yêu thích, đam mê với lập trình thi đấu, ngoài ra còn mang đến cho mình cả một chân trời kiến thức mới, hầu hết kiến thức mà mình học được đến từ các dự án VNOI Wiki và VNOJ. Contest online về CP đầu tiên mà mình làm là Bedao Mini Contest 11, mình khá ấn tượng về cách chấm bài online nhanh gọn trên VNOJ, khác với việc mình phải đợi rất lâu để được các thầy dạy đội tuyển hồi cấp 2 của mình chấm bằng Themis, trải nghiệm hoàn toàn mới này khiến mình có thêm động lực, hứng thú để tiếp tục làm các contest tiếp theo. Qua series contest Bedao, bản thân mình cảm thấy đã tiến bộ rất nhiều, lúc ấy

mình cũng mong muốn có thể một ngày nào đó có thể tổ chức những contest như thế này. Và rồi mình biết đến khái niệm 'Tình nguyện viên VNOI' là những người đứng sau các contest nói trên và VNOI Wiki qua bài tuyển TNV Gen 3 trên page VNOI, từ đó mình tìm hiểu và nuôi hi vọng có thể trở thành TNV VNOI trong tương lai.

Một trong những lý do chính thúc đẩy mình trở thành một TNV VNOI đó chính là mong muốn được đóng góp cho cộng đồng. Các dự án của VNOI đã giúp mình rất nhiều trên con đường theo đuổi lập trình thi đấu của mình, giúp mình tiến bộ hơn và đạt được những điều mà mình tưởng chừng như không thể, do đó mình cũng mong muốn có thể cùng VNOI tạo thêm nhiều giá trị tốt đẹp cho mọi người, nhất là những bạn có niềm yêu thích với tin học. Qua việc trở thành một tình nguyện viên, mình cũng mong muốn phát triển bản thân mình hơn, rèn dũa thêm kiến thức và kỹ năng mềm, đồng thời tạo thêm nhiều sự kết nối và mối quan hệ với những người có cùng niềm yêu thích với mình, mình nghĩ quá trình giao lưu và học hỏi này sẽ giúp bản thân mình ngày càng trở thành phiên bản tốt nhất. Một lý do khác là vì mình thấy icon 'lửa' ở tên dành cho TNV trên VNOJ khá ngẫu, và cách duy nhất để đạt được là trở thành một TNV VNOI. Mình cũng thích các sản phẩm, merch của VNOI như áo, sticker hay móc khóa, vì thế trở thành TNV cũng phần nào hiện thực hóa được ước mơ nhỏ nhoi này của mình, tạo thêm cơ hội để 'flex' với bạn bè. Ngoài ra, làm TNV VNOI, mình còn được tham gia các buổi talkshow, workshop với các gương mặt nổi tiếng của Tin học Việt Nam, được giao lưu, học hỏi với nhiều người giỏi, trau dồi các kỹ năng mềm và nhiều nhiều lợi ích khác nữa.



Các bạn TNV cùng nhau chơi Boardgame tại buổi sinh hoạt

Do đã đặt mục tiêu và quyết tâm về việc trở thành TNV VNOI từ trước, khi thời điểm tháng

⁷CP: Competitive Programming - Lập trình thi đấu



8 cận kề, hầu như ngày nào mình cũng vào page VNOI để xem tin tức về việc tuyển thành viên, và khi bài viết ấy được đăng lên - thứ mà mình đã mong chờ bấy lâu, mình biết thời khắc ấy đã đến. Đọc từng dòng trong file mô tả công việc, mình cảm thấy yêu thích và hứng thú với việc trở thành thành viên của ban contest, vì mình không giỏi về mảng kĩ thuật hay đủ khả năng viết để viết tài liệu wiki, một phần khác vì đây là ban tuyển nhiều thành viên nhất, nên mình nghĩ khả năng đậu sẽ cao hơn. Ngoài ra, khi làm thành viên ban contest, mình có thể trau chuốt thêm về kĩ năng và kiến thức lập trình thi đấu của mình, cũng như hiểu được cách tạo một problem hay hơn là quy trình tổ chức một contest, đồng thời mình cũng mong muốn có thể đóng góp thêm những bài toán hay ho trong kho bài tập của VNOI. Bắt tay vào việc điền vào đơn tuyển, mình cố gắng hết sức để lá đơn trở nên thật chín chu và hoàn hảo nhất. Các câu hỏi trong đơn tuyển tuy không nhiều và không quá dài, nhưng làm ngốn thời gian của mình rất nhiều. Mỗi ngày mình đều bật máy lên để xem lại các thông tin đã điền để thêm thắt và chỉnh sửa, với mục đích để lá đơn của mình nổi bật nhất có thể do mình biết còn rất nhiều bạn cũng đang mong ước được làm một phần của team VNOI như mình. Rồi cũng đến lúc đóng hạn nộp đơn, mình review và chỉnh sửa vài lần trước khi gửi. Sau đó đơn tuyển được gia hạn thêm vài tuần, vì đã soạn sẵn từ trước nên trong 2 tuần mình cũng không chỉnh sửa gì nhiều. Quãng thời gian chờ đợi kết quả, ngày nào mình cũng ngồi đọc các tin nhắn trong server discord của VNOI để xem có ai bàn luận về lần tuyển TNV này không, mình check mail và mở page VNOI trên facebook thường xuyên hơn, với mong muốn nhận được mail thông báo kết quả đợt tuyển càng sớm càng tốt. Ngày nhận được kết quả chính thức trở thành một TNV, mình rất vui nhưng cũng đan xen hồi hộp, vui vì mình đã hoàn thành được mục tiêu của mình, hồi hộp và lo lắng vì không biết nhiệm kì sắp tới của mình sẽ trải qua những gì, sẽ gặp những ai, sẽ học được điều gì mới.

Hành trình trở thành tình nguyện viên VNOI có lẽ sẽ là một trong những quãng thời gian đáng nhớ nhất của mình, vì có lẽ mình của trước đây cũng không thể nghĩ rằng mình có thể trở thành một phần của VNOI, do đó việc trở thành TNV mang lại ý nghĩa rất to lớn với mình. Dù chỉ mới trải qua một nửa nhiệm kì Gen 4, nhưng mình cũng đã cảm nhận được tinh thần và sự tận tụy trong công việc của các thành viên khác trong khi bản thân vẫn chưa đóng góp được nhiều, từ đó tạo thêm động

lực trong mình để có thể tiếp tục cố gắng góp phần giúp cộng đồng lập trình thi đấu ở Việt Nam ngày một lớn mạnh hơn nữa."

VNOI CUP 2024 - Cuộc đua trí tuệ không giới hạn

VNOI CUP 2024, một trong những sự kiện lập trình được mong đợi nhất trong năm được tổ chức bởi VNOI, đã khép lại với những thành công rực rỡ. Lần thứ ba cuộc thi được tổ chức, đây không chỉ là sân chơi cho các bạn trẻ yêu thích lập trình thi đấu thể hiện tài năng mà còn là cầu nối để cộng đồng lập trình Việt Nam giao lưu, học hỏi và phát triển. Bài viết này sẽ đưa quý độc giả đi qua hành trình đầy kịch tính của VNOI CUP 2024, từ những vòng loại sôi động đến chung kết đầy căng thẳng.

Tổng quan về VNOI CUP 2024

Năm 2024, VNOI CUP đã đánh dấu một bước tiến mới với nhiều đổi mới về thể lệ, quy mô và chất lượng đề thi. Điểm đáng chú ý là VNOI CUP 2024 không giới hạn độ tuổi tham dự để tạo điều kiện cho mọi thí sinh ở mọi độ tuổi đều có thể tham gia; cùng với đó, số lượng thí sinh được tham dự Vòng Chung kết tăng lên gấp đôi so với năm 2023, thành 24 thí sinh.

Hai vòng loại diễn ra sôi nổi với sự tham gia lên đến 1900 thí sinh trong một vòng thi, là số thí sinh tham gia kỷ lục trong tất cả các kì thi đã từng tổ chức trên hệ thống VNOJ. Vòng Chung kết diễn ra tại Trường Đại học Khoa học tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh đã thu hút sự quan tâm của đông đảo cộng đồng lập trình thi đấu. Các thí sinh xuất sắc nhất đã có cơ hội giao lưu, học hỏi và cùng nhau tranh tài trong một không khí vô cùng sôi động.



Timeline VNOI CUP 2024

Ban ra đề VNOI CUP 2024



Thành tích ban ra đề VNOI CUP 2024

Để cho "ra lò" các bài tập đầy thử thách, đòi hỏi sự miệt mài và sáng tạo rất cao trong khâu chuẩn bị và ra đề. Bên cạnh những gương mặt thân quen trong ban ra đề từ các cuộc thi trước, VNOI CUP 2025 cũng đã chào đón thêm nhiều thành viên tài năng mới.

- Đặng Đoàn Đức Trung: Trưởng ban ra đề VNOI CUP 2023 & 2024 - ICPC⁸ World Finalist 2023, 2022, 2021, 2020; Max 2871 Codeforces⁹.
- Trần Quang Lộc: Trưởng ban ra đề ICPC Asia HCMC 2022, ICPC Asia Hue City 2023; Coordinator¹⁰ Codeforces; Max 2546 Codeforces.
- Nguyễn Đình Quang Minh: ICPC World Finalist 2018, 2021; Max 2726 Codeforces.
- Hồ Ngọc Vĩnh Phát: ICPC World Finalist 2023, 2022; HCB IOI¹¹ 2021; HCV APIO¹² 2021; Max 2345 Codeforces.
- Nguyễn Diệp Xuân Quang: ICPC World Finalist 2020, 2021; HCB APIO 2017, Max 2446 Codeforces.

⁸ICPC: International Collegiate Programming Contest - Cuộc thi Lập trình Quốc tế lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu.

⁹Codeforces: một trang web tổ chức các cuộc thi lập trình thi đấu với các dạng đề, bài tập đa dạng.

¹⁰Coordinator: người tổ chức các contest cho OJ.

¹¹IOI: Kỳ thi Olympic Tin học Quốc tế.

¹²APIO: Kỳ thi Olympic Tin học Châu Á — Thái Bình Dương.

¹³HSGQG: là viết tắt của kỳ thi Học sinh giỏi Quốc gia.



- Nguyễn Hoàng Vũ: ICPC World Finalist 2023; HCB IOI 2021; Max 2204 Codeforces.
- Phạm Xuân Trung: VNOI CUP 2022 Finalist; giải Nhất HSGQG¹³ 2022; Max 2259 Codeforces.
- Hoàng Quốc Việt: giải Nhì HSGQG 2022; Max 2139 Codeforces.
- Vũ Hoàng Kiên: Bronze Medal ICPC World Finals¹⁴ 2021, ICPC World Finalist 2023, HCB IOI 2019, 2020; HCB APIO 2019, 2020.
- Trương Văn Quốc Bảo: HCB IOI 2021, 2022; HCV APIO 2021, 2022.
- Lê Quang Huy: HCB IOI 2020, 2021; HCV APIO 2021; HCB APIO 2020.

số hai chỉ hơn 3 phút thời gian làm bài mà thôi.

Danh sách thí sinh dự thi Vòng Chung kết

duongnc000	noobcoder	chemthan	khanhphucscratch
fextivity	6aren	flashmt	toib
gawr_gura	hollwo_pelw	luanlinh17032006	tungvh04
marvinthang	Yurushia	kilkuwu	Pannda
minhcool	LeKienThanh69	hoangxuanbach	DeMen100ns
socpita	neko_nyaaaaaaaaa	cuom1999	powder

Danh sách thí sinh tham dự VCK VNOI CUP 2024

Vòng loại: Bước đệm vững chắc

Vòng loại của VNOI CUP 2024 đã diễn ra vô cùng sôi động với sự tham gia của hàng ngàn thí sinh tài năng. Cuộc đua giành vé vào Vòng Chung kết đã trở nên vô cùng hấp dẫn và gay cấn. Cùng với đó, việc thay đổi số lượng vòng loại VNOI CUP năm nay đòi hỏi hệ thống các bài trong kì thi có sự liên kết và tính phân loại cao để đánh giá đúng thực lực của thí sinh dự thi nhằm chọn ra những thí sinh tài năng nhất cho Vòng Chung kết VNOI CUP 2024. Chính vì vậy, các bài tập trong vòng loại năm nay cũng có phần thử thách hơn những năm trước đó.

Ở vòng loại đầu tiên, ban tổ chức đã ghi nhận con số kỷ lục lên đến gần 1900 thí sinh dự thi, phá kỷ lục 1369 thí sinh của VNOI CUP 2023. Với phong độ thi đấu vô cùng chắc chắn, bạn SanguineChameleon đã rất xuất sắc AC được 6/7 bài tập đầu tiên của kỳ thi và giành ngôi vị dẫn đầu, trong đó các bài tập đều chỉ được nộp một lần duy nhất. Ở vị trí thứ hai và ba lần lượt là bạn Nguyễn Vũ Đăng Huy (Yurushia) và bạn Đỗ Đình Dắc (6aren).

Tại vòng loại thứ hai, cuộc thi diễn ra không kém phần căng thẳng. Với kinh nghiệm chinh chiến từ nhiều cuộc thi lớn trong và ngoài nước, Trần Xuân Bách (fextivity) đã xuất sắc dẫn đầu vòng thi với số điểm 12250/15250, AC 6/7 bài toán. Xếp thứ hai và ba lần lượt là Phạm Quốc Hùng (hollwo_pelw) và Nguyễn Đức Thắng (marvinthang), cũng đồng số điểm 12250/15250. Đáng nói, khoảng cách giữa vị trí dẫn đầu và vị trí

Sau hai vòng thi loại cùng với đề cử từ cộng đồng, 24 cái tên xuất sắc góp mặt tại Vòng Chung kết đã lộ diện:

- Nguyễn Tấn Sỹ Nguyên - flashmt (30 tuổi), đến từ TP.HCM, Hạng 13 tổng điểm 2 vòng loại.
- Nguyễn Xuân Tùng - neko_nyaaaaaaaaaaaaa.. (25 tuổi), đến từ TP.HCM, Hạng 11 tổng điểm 2 vòng loại.
- Võ Khắc Triệu - DeMen100ns (19 tuổi), đến từ TP.HCM, Thí sinh xuất sắc do BTC đề cử.
- Đỗ Đình Dắc - 6aren (25 tuổi), đến từ Hà Nội, Hạng 3 vòng loại thứ nhất.
- Lê Đức Toàn - toib (20 tuổi), đến từ Biên Hòa, Đồng Nai, Hạng 19 tổng điểm 2 vòng loại.
- Lê Kiên Thành - LeKienThanh69 (17 tuổi), đến từ Quy Nhơn, Bình Định, Hạng 10 tổng điểm 2 vòng loại.
- Lại Minh Quang - gawr_gura (18 tuổi), đến từ Thanh Hoá, Hạng 3 tổng điểm 2 vòng loại.
- Nguyễn Đức Thắng - marvinthang (18 tuổi), đến từ Phú Thọ, Hạng 3 vòng loại thứ hai.
- Hà Lộc Hải - powder (19 tuổi), đến từ Thanh Hoá, Thí sinh xuất sắc do BTC đề cử.
- Nguyễn Quang Minh - minhcool (19 tuổi), đến từ Hà Nội, Hạng 6 tổng điểm 2 vòng loại.
- Lê Phước Định - cuom1999 (25 tuổi), đến từ Đà Nẵng, Hạng 17 tổng điểm 2 vòng loại.
- Lê Nguyễn Hữu An - Pannda (18 tuổi), đến từ TP.HCM, Hạng 21 tổng điểm 2 vòng loại.
- Nguyễn Khánh Phúc - khanhphucscratch (14 tuổi), đến từ Ninh Bình, Hạng 18 tổng điểm

¹⁴World Finals: Cuộc thi ICPC cuối cùng trong một mùa giải bao gồm những đội xuất sắc nhất của các trường đại học trên thế giới vượt qua vòng loại vùng (Regional contest)



2 vòng loại.

- Vũ Hoàng Tùng - tungvh04 (20 tuổi), đến từ TP.HCM, Hạng 20 tổng điểm 2 vòng loại.
- Nguyễn Tuấn Linh - tuanlinh17032006 (18 tuổi), đến từ Hà Nội, Hạng 22 tổng điểm 2 vòng loại.
- Phạm Công Minh - socpите (18 tuổi), đến từ Hà Nội, Hạng 7 tổng điểm 2 vòng loại.
- Phạm Quốc Hùng - hollwo_pelw (19 tuổi), đến từ Hà Nội, Hạng 2 vòng loại thứ hai.
- Lê Hoàng Nam - noobcoder (18 tuổi), đến từ Hà Nội, Hạng 8 tổng điểm 2 vòng loại.
- Trần Xuân Bách - festivity (19 tuổi), đến từ Hà Nội, Hạng 1 vòng loại thứ hai.
- Nguyễn Tùng Lâm - kilkuwu (18 tuổi), đến từ Hải Phòng, Hạng 15 tổng điểm 2 vòng loại.
- Nguyễn Vũ Đăng Huy - Yurushia (21 tuổi), đến từ TP.HCM, Hạng 2 vòng loại thứ nhất.
- Nguyễn Cảnh Dương - duongnc000 (17 tuổi), đến từ Hà Nội, Hạng 1 tổng điểm 2 vòng loại.
- Hoàng Xuân Bách - hoangxuanbach (17 tuổi), đến từ Hà Nội, Hạng 16 tổng điểm 2 vòng loại.
- Nguyễn Ngọc Trung - chemthan (32 tuổi), đến từ Phú Thọ, Hạng 12 tổng điểm 2 vòng loại.

Những cái tên vô cùng xuất sắc kể trên đã tạo nên một Vòng Chung kết kịch tính và khó quên tại VNOI CUP 2024.

Vòng Chung kết: Đỉnh cao của tài năng



Vòng chung kết VNOI CUP 2024 được phối hợp tổ chức tại TP.HCM từ ngày 01/08 - 04/08

Vòng Chung kết VNOI CUP 2024 đã diễn ra tại Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh. 24 thí sinh xuất sắc nhất từ vòng loại đã hội tụ tại đây để tranh tài.

Đáng nói, độ tuổi của các thí sinh tham dự Vòng Chung kết rất đa dạng, trong đó có bạn

Nguyễn Khánh Phúc (khanhphucscratch) chỉ mới 14 tuổi. Cùng với đó, danh sách này cũng có sự góp mặt của 2/4 thành viên đội tuyển IOI Việt Nam 2024, đó là Hoàng Xuân Bách (hoangxuanbach) và Phạm Công Minh (socpите); đây cũng chính là hai gương mặt vàng đã xuất sắc mang về hai tấm huy chương vàng cho đoàn Việt Nam.

Giống như các cuộc thi trước, Vòng Chung kết VNOI CUP 2024 tiếp tục mở thêm bảng mở rộng - bảng đấu dành cho mọi thí sinh mong muốn có cơ hội cọ sát trực tiếp tại địa điểm thi của các thí sinh chính thức.

Các bài trong Vòng Chung kết sử dụng những kiến thức đã khá quen thuộc với cộng đồng Tin học Việt Nam trong những năm gần đây, có độ phân hoá cao. Ngoài ra, thí sinh năm nay trải dài ở nhiều thế hệ, do đó, cuộc thi đã diễn ra rất căng thẳng và khó đoán định trước được kết quả.



Không khí VCK VNOI CUP 2024 tại trường Đại học Khoa học tự nhiên, TP.HCM

Trải qua hàng giờ đồng hồ căng thẳng với không khí không thể nào rực lửa hơn của Vòng Chung kết, trong một khoảnh khắc, contest đã chứng kiến sự vượt lên của bạn Lê Hoàng Nam (noobcoder) - thí sinh giải được bài E sớm nhất, trước khi các đối thủ bám đuổi phía sau nhanh chóng bắt kịp sau 2,5 tiếng tranh tài. Nửa sau của contest, Trần Xuân Bách (festivity) thể hiện đẳng cấp vượt trội của mình khi là thí sinh duy nhất chinh phục bài G trị giá 3500 điểm, qua đó tạo cách biệt lớn với vị trí thứ hai và trở thành nhà Vô địch VNOI CUP 2024 với màn trình diễn thuyết phục. Chung cuộc, ba vị trí đầu tiên của Vòng Chung kết VNOI Cup 2024 đã thuộc về ba cái tên:

- Trần Xuân Bách (19 tuổi) - Nhà Vô địch VNOI CUP 2024 với số điểm 14750, nhận được phần thưởng trị giá 20 triệu VNĐ.
- Lê Phước Định (25 tuổi) - Runner-up VNOI CUP 2024 với số điểm 10250, nhận được phần thưởng trị giá 15 triệu VNĐ.



- Nguyễn Tấn Sỹ Nguyên (30 tuổi) - Giành cup đồng VNOI CUP 2024 với số điểm 10000, nhận được phần thưởng trị giá 10 triệu VNĐ.



Chân dung TOP 3 VCK VNOI CUP 2024: Lê Phước Định, Trần Xuân Bách, Nguyễn Tấn Sỹ Nguyên

Lời kết

Hai tháng - 24 thí sinh - 1 Nhà Vô Địch, VNOI CUP 2024 đã đi đến hồi kết với một kì chung kết cực kì mãn nhãn. Cuộc thi may mắn được chứng kiến những màn trình diễn cực kì thuyết phục từ tất cả thí sinh tham gia dự thi; bao gồm cả những màn khăng định sức mạnh đến từ các gương mặt quen thuộc, và sự tiến bộ rất đáng mong chờ mà các thế hệ "măng non" của nền Tin học Việt Nam có thể đem lại.

Như vậy, VNOI CUP 2024 đã khép lại thành công tốt đẹp. Cuộc thi không chỉ là một sân chơi cho các bạn trẻ yêu thích lập trình mà còn là một cầu nối để các tài năng trẻ Việt Nam được giao lưu, học hỏi và phát triển. Kỳ thi đã khép lại với vô vàn cảm xúc khác nhau, nhưng đọng lại vẫn là những cảm xúc không thể nào quên được. Cánh cửa này khép lại để chờ đón những cánh cửa khác mở ra, hãy cùng nhau hướng đến VNOI CUP 2025 nhé!



Tập thể BTC và đại diện trường Đại học Khoa học tự nhiên, TP.HCM cùng các thí sinh của VCK VNOI CUP 2024

OLP'24 & ICPC Asia Hanoi 2024 - Nhiều dấu ấn khó quên

Năm 2024, cộng đồng lập trình thi đấu Việt Nam lại một lần nữa được chứng kiến những màn trình diễn đỉnh cao tại các kỳ thi ICPC và Olympic Tin học Sinh viên. Cùng nhìn lại các kỳ thi này và hành trình VNOI đồng hành tại các kỳ thi nhé!

ICPC¹⁵ Asia Pacific Championship 2024

Vào cuối năm 2023, khu vực Châu Á - Thái Bình Dương đã tổ chức 5 điểm thi ICPC Asia Pacific Regional tại: Taoyuan - 23/10 (Đài Loan), Seoul - 25/11 (Hàn Quốc), Yokohama - 26/11 (Nhật Bản), Jakarta - 04/12 (Indonesia), Hue City - 08/12 (Việt Nam). Nối tiếp những thành công của ICPC Asia Hue City 2023, nhằm chọn ra những đội thi xuất sắc đại diện Asia Pacific tham dự Chung kết ICPC toàn cầu năm 2024 tại Kazakhstan, các thí sinh phải cạnh tranh Ngôi vị Vô địch và 16 suất tham dự ICPC World Finals¹⁶ qua Chung kết ICPC Asia Pacific. Với thể thức mới được áp dụng từ kỳ ICPC 2024, Hà Nội đã chính thức trở thành địa điểm đầu tiên được đăng cai tổ chức Chung kết ICPC Asia Pacific 2024 phối hợp với ICPC Việt Nam và ICPC Asia Pacific.



GS.TS Chủ Đức Trình, Hiệu trưởng Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, Trưởng ban tổ chức phát biểu

Kỳ thi diễn ra từ ngày 29/02 - 03/03/2024 tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội. Kỳ thi đã được tổ chức trực tiếp thi đấu trong 5 tiếng từ 9h00 - 14h00 ngày 02/03/2024 tại Khu thi ICPC tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội. Kỳ thi được diễn ra trên Hệ thống

chuẩn Quốc tế ICPC được ICPC Toàn cầu và Asia Pacific chỉ định và vận hành, phần mềm giám sát thi đấu của Việt Nam do VNOI vận hành.

Kỳ thi quy tụ 65 đội thi đến từ các trường đại học hàng đầu của Hàn Quốc, Nhật Bản, Indonesia, Singapore, Việt Nam,... Trong đó, Việt Nam góp mặt 12 đại diện đến từ cả ba miền:

- Đại học Bách khoa Hà Nội: HUST Kite;
- Đại học Công nghệ, Đại học Quốc gia Hà Nội: sudo, cat, rmdir;
- Đại học FPT Hà Nội: FPTU Ching Cheng Hanji, FPTU Right Triangle;
- Đại học Bách khoa Đà Nẵng: BKDN.LeaderTurnsBack;
- Đại học Khoa học tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh: HCMUS-SmokinTomatoes, HCMUS-ExploringWorld, HCMUS-PeTaINyKing;
- Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh: UIT.HHQ.SilverBullet;
- Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh: HCMUTE.Fortune.



Bên trong địa điểm tổ chức thi của Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

Sau 5 tiếng thi đấu căng thẳng, ngôi vô địch ICPC Asia Pacific Championship 2024 đã thuộc về đội tuyển NewTrend đến từ Đại học Quốc gia Seoul

¹⁵ICPC: International Collegiate Programming Contest - Cuộc thi Lập trình Quốc tế lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu.

¹⁶World Finals: Cuộc thi ICPC cuối cùng trong một mùa giải bao gồm những đội xuất sắc nhất của các trường đại học trên thế giới vượt qua vòng loại vùng (Regional contest)

(Hàn Quốc). Từ kết quả kỳ thi, Việt Nam có ba đại diện xuất sắc giành được tấm vé tham dự ICPC World Finals 2024, chính là:

- UET sudo (Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội): Phạm Quốc Hùng, Hoàng Ngọc Bảo Khuê, Phạm Xuân Trung
- HUST Kite (Trường Đại học Bách khoa Hà Nội): Nguyễn Anh Tiến, Phạm Hoàng Tiến, Nguyễn Tấn Dũng
- HCMUS-ExploringWorld (Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh): Lê Hữu Nghĩa, Lê Minh Hoàng, Vũ Hoàng Tùng

Olympic Tin học Sinh viên Việt Nam và ICPC Asia Hanoi 2024



PGS.TS. Nhà giáo nhân dân Hồ Sĩ Đàm chủ tịch hội đồng giám khảo

Kỳ thi Olympic Tin học Sinh viên Việt Nam lần thứ 33, kỳ thi PROCON Việt Nam và kỳ thi Lập trình sinh viên Quốc tế ICPC Asia Hanoi 2024 được tổ chức tại Đại học Công nghiệp Hà Nội từ ngày 10 - 13/12/2024. Kỳ thi quy tụ gần 700 coders xuất sắc nhất đến từ 78 trường đại học, cao đẳng, học viện trên khắp cả nước; cùng với đó là 20 đội tuyển ICPC Quốc tế tài năng đến từ Singapore, Hàn Quốc, Indonesia, Thái Lan, Philippines và Đài Loan. Kỳ thi năm 2024 có quy mô lớn nhất từ trước đến nay với hơn 700 thí sinh và 180 huấn luyện viên đến từ 116 trường đại học, cao đẳng, học viện trên cả nước và các quốc gia Singapore, Hàn Quốc, Đài Loan, Thái Lan.



Nguyễn Đức Thắng (Trường Đại học Kinh tế Quốc dân Hà Nội) - Vô địch Siêu CUP OLP'24.

Đối với kỳ thi Olympic Tin học Sinh viên Việt Nam, các vị trí dẫn đầu mỗi khối thi như sau:

- Vô địch khối Siêu Cúp: bạn Nguyễn Đức Thắng - Đại học Kinh tế Quốc dân
- Vô địch khối Chuyên Tin: bạn Đỗ Anh Khoa - Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh
- Vô địch khối Không Chuyên & Cao Đẳng: bạn Đinh Văn Tùng - Trường Đại học Thủy Lợi

Giải đồng đội khối Chuyên Tin và Không Chuyên của kỳ thi lần lượt thuộc về Đại học Bách khoa Hà Nội và Đại học Kinh tế Quốc dân. Như thông lệ, ngoài ra ban tổ chức còn trao thêm giải nữ sinh xuất sắc nhất dành cho các "bông hồng" đã tham gia kỳ thi. Cụ thể, bạn Nguyễn Thanh Tâm - Trường Đại học Ngoại Thương, Cơ sở II tại Thành phố Hồ Chí Minh với giải Ba khối Không Chuyên cùng với bạn Trần Hương Ly - Trường Đại học Ngoại Thương với giải Nhất khối Chuyên Tin, đã xuất sắc được trao giải thưởng này.



Việt Nam có 2 đội tuyển giành Huy chương Vàng tại ICPC Asia Hanoi 2024.

Còn đối với kỳ thi Lập trình sinh viên Quốc tế ICPC Asia Hanoi 2024, ngôi vô địch thuộc về đội tuyển fruit_advantages đến từ Đại học Quốc gia Đài Loan, sau màn bức phá AC thành công bài I ở phút thứ 290, nâng tổng số bài giải được lên 11/13 bài. Riêng Việt Nam có hai đội tuyển

xuất sắc giành được huy chương vàng là HCMUS-AtCoder (Trường Đại học Khoa học tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh) và UIT.KHQ.TheUnkillableDemonKings (Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh). Kết quả của vòng thi này là cơ sở để lựa chọn ra các đội tuyển xuất sắc tranh tài ở ICPC Asia Pacific Championship 2025 tại Singapore vào tháng 03/2025.



Ngôi Vô địch ICPC Asia Hanoi 2024 thuộc về đội tuyển fruit_advantages (Đại học Quốc gia Đà Loan)

VNOI đồng hành cùng Olympic Tin học Sinh viên Việt Nam



Những thành viên VNOI đồng hành cùng các thí sinh tại ICPC Asia Hanoi

Năm 2024 đánh dấu năm thứ tư VNOI đồng hành cùng các kỳ thi ICPC và Olympic Tin học Sinh viên. Kế thừa và phát triển công nghệ, kỹ thuật và kinh nghiệm từ các năm trước, VNOI tiếp tục hỗ trợ tổ chức các sự kiện diễn ra thuận lợi, tốt đẹp.

Trong các kỳ thi kể trên, VNOI rất vinh dự và tự hào khi hệ thống chấm thi cũng như hệ thống giám sát thi do chính VNOI thiết kế dựa trên bản máy thi của ICPC World Finals được đưa vào sử dụng. Các hệ thống đều nhận được nhiều phản hồi

tích cực từ ban tổ chức và các thí sinh dự thi, cả thí sinh trong nước lẫn thí sinh quốc tế. Riêng đối với hệ thống máy ảo thi, so với lần đầu sử dụng tại kỳ thi ICPC Regional Hue City 2023, đến nay hệ thống đã được cập nhật và hiệu chỉnh nhiều tính năng, góp phần giúp thí sinh có những trải nghiệm làm bài tốt nhất. Đặc biệt, với cải tiến ở tính năng in ấn mã nguồn, chỉ mất một thời gian rất ngắn để bản in tới được tay thí sinh từ khi yêu cầu được gửi đi, góp phần nâng cao hiệu quả thi đấu của các đội tuyển ICPC.

Lời kết

Năm 2024 là một năm thành công của cộng đồng lập trình Việt Nam với những thành tích đáng nể tại các đấu trường quốc tế. Đây chính là minh chứng rõ nét cho tài năng và sự nỗ lực không ngừng của các bạn sinh viên, của thế hệ các bạn trẻ yêu thích lập trình.

VNOI, với vai trò là cầu nối giữa các lập trình viên trẻ và các cuộc thi quốc tế, đã đóng góp không nhỏ vào sự phát triển của cộng đồng lập trình Việt Nam. Hệ thống chấm thi và giám sát thi do VNOI xây dựng đã đảm bảo tính công bằng và minh bạch cho các kỳ thi, đồng thời tạo điều kiện thuận lợi cho các thí sinh thể hiện hết khả năng của mình.

Những thành công này là nguồn động viên lớn cho thế hệ trẻ Việt Nam, khẳng định vị thế của Việt Nam trên bản đồ lập trình thế giới. Có thể nói, các cuộc thi như ICPC và Olympic Tin học Sinh viên Việt Nam không chỉ là nơi để các bạn sinh viên thể hiện tài năng mà còn là nơi để các bạn trẻ học hỏi, giao lưu và cùng nhau phát triển. Trong năm tới và trong tương lai, VNOI vẫn sẽ tiếp tục đồng hành và hỗ trợ cộng đồng lập trình Việt Nam phát triển, qua việc tổ chức các hoạt động và duy trì các cuộc thi chất lượng cao.



Thuật toán tìm số nguyên tố của Dijkstra

*Võ Đức Đoàn
11 Tin, Trường THPT Chuyên Nguyễn Tất Thành*



Edsger W. Dijkstra là một nhà khoa học máy tính có nhiều đóng góp trong lĩnh vực ngôn ngữ lập trình và khoa học máy tính, nổi tiếng nhất chính là thuật toán Dijkstra - một thuật toán tìm đường đi ngắn nhất, được ông nghĩ ra trong vòng 20 phút khi đi cafe cùng vị hôn thê của mình.

Thế nhưng, bạn có biết rằng Dijkstra cũng phát minh ra một thuật toán tìm số nguyên tố không?

Số nguyên tố

Nhắc lại định nghĩa Số nguyên tố là một số tự nhiên có đúng hai ước nguyên dương 1 và chính nó. Các số tự nhiên còn lại được gọi là hợp số.

Ví dụ:

- 7 là số nguyên tố vì nó chỉ có 2 ước là 1 và 7
- 12 không là số nguyên tố vì nó có 6 ước là 1, 2, 3, 4, 6 và 12

Vì các số nguyên tố phân bố theo một quy luật không xác định nên việc xác định các số nguyên tố trong một khoảng số là một thử thách lớn. Các nhà toán học thích sự thử thách, và giờ đây ta có vô số cách xác định số nguyên tố.

Ta có bài toán:

Cho một số tự nhiên N . Hãy tìm và in ra tất cả các số nguyên tố từ 1 đến N .

Trước khi giới thiệu về cách giải quyết bài toán này bằng thuật toán tìm số nguyên tố của Dijkstra, ta sẽ đi qua một số thuật toán tìm số nguyên tố nhằm giải quyết được bài toán trên.

Phương pháp thử sai

Một trong những cách đơn giản nhất để kiểm tra một số x có phải là một số nguyên tố không chính là xét tất cả các số từ 2 đến $x - 1$ và kiểm tra nếu x có chia hết cho số nào trong khoảng này không. Nếu như không tồn tại số nào thỏa mãn thì x số nguyên tố, không thì ngược lại.

```
bool isPrime(int x) {
    if (x <= 1) return 0; // số 1 không phải số nguyên tố
    for (int i = 2; i < x; ++i)
        if (x % i == 0) return 0;
    return 1;
}
```

Độ phức tạp của thuật toán này sẽ là $O(n)$ với $n = x$.

Ta có thể giảm bớt số lượng thao tác cần thực hiện bằng cách chỉ xét các số nằm trong khoảng từ 2 đến $\sqrt{x}$.

Tại sao chỉ cần xét đến $\sqrt{x}$? Ví dụ với số 12. Ta có $\sqrt{12} > 3$ và các ước của 12 nhỏ hơn $\sqrt{12}$ chính là các số 1, 2 và 3. Dễ thấy, khi lấy 12 chia cho các ước này, ta dễ dàng tìm được các ước số còn lại của 12:

- $4 = \frac{12}{3}$
- $6 = \frac{12}{2}$
- $12 = \frac{12}{1}$

Khi này, độ phức tạp của phương pháp thử sai được giảm xuống còn $O(\sqrt{n})$.

```
bool isPrime(int x) {
    if (x <= 1) return 0;
    for (int i = 2; i * i <= x; ++i) { // i <= sqrt(x) <=> i * i <= x
        if (x % i == 0) return 0;
    }
    return 1;
}
```

Sau khi đã có thuật toán xác định x có phải số nguyên tố không, ta có thể giải quyết bài toán trên: Ta xét các số từ 1 đến n , nếu số đó là số nguyên tố, ta in số đó ra.

```
for (int i = 1; i <= n; ++i)
    if (isPrime(i)) cout << i << '\n';
```

Độ phức tạp của chương trình sẽ là $O(n\sqrt{n})$.

Vì chương trình của ta không cần lưu bất kì thông tin gì thêm nên độ phức tạp bộ nhớ sẽ là $O(1)$.

Tối ưu

Nhìn lại bài toán, ta nhận thấy chương trình có thể được tối ưu hơn nữa cho dạng bài này bằng cách tận dụng các số nguyên tố tìm được những bước trước đó để kiểm tra số x .

```
vector<int> primes;

bool isPrime(int x) {
    if (x <= 1) return 0;
    for (int i : primes) {
        if (i * i > x) break;
        if (x % i == 0) return 0;
    }
    return 1;
}

for (int i = 1; i <= n; ++i) {
    if (isPrime(i)) {
        cout << i << '\n';
        primes.push_back(i);
    }
}
```



Thuật toán bây giờ có độ phức tạp $O(n \times \pi(n))$, với $\pi(n)$ là số lượng số nguyên tố từ 1 đến n . Theo [định lý số nguyên tố](#) ¹⁷, ta có: $\pi(n) \approx \frac{n}{\log_e n}$, nên độ phức tạp trên sẽ tương đương: $O(n \times \pi(n)) \approx O(n \frac{n}{\log_e n})$.

Độ phức tạp bộ nhớ bây giờ tăng lên thành $O(\pi(n)) \approx O(\frac{n}{\log_e n})$ vì chương trình giờ đây phải lưu thêm danh sách các số nguyên tố từ 1 đến n .

Thuật toán sàng số nguyên tố

Thuật toán sàng số nguyên tố cho phép ta xác định các số nguyên tố trong khoảng từ 1 đến N với độ phức tạp nhanh nhất ngưỡng: $O(n \log \log n)$.

Thuật toán được thực hiện dựa trên nhận định sau: nếu p là một số nguyên tố thì các bội số khác p của p như $2p, 3p, 4p, \dots$ không phải là một số nguyên tố.

Từ đây ta có mô tả thuật toán:

Ta có một mảng đánh dấu `notPrime` có n phần tử được đánh số từ 1. Ban đầu, `notPrime[1] = 1` vì 1 không phải là số nguyên tố.

Ta duyệt các số từ 2 đến n , với mỗi số x được duyệt, ta có 2 trường hợp:

- `notPrime[x] = 0`: số x không được đánh dấu $\rightarrow x$ là số nguyên tố. Khi này, bắt đầu từ bội số thứ x của x , ta đánh dấu `notprime[x*x] = 1`, `notprime[(x + 1)*x] = 1`, ..., cho tới `notprime[(n/x)*x] = 1`.
- `notPrime[x] = 1`: số x được đánh dấu nên x không phải số nguyên tố. Khi này ta không làm gì cả.

Sau khi kết thúc thuật toán, vị trí của các `notPrime` không được đánh dấu sẽ là một số nguyên tố.

```
const int N = 1e6 + 10; // N > n
bool notPrime[N];
for (int i = 2; i <= n; ++i) {
    if (notPrime[i] == 0)
        for (int j = i * i; j <= n; j += i)
            notPrime[j] = 1;
}
for (int i = 2; i <= n; ++i)
    if (notPrime[i] == 0) cout << i << '\n';
```

Câu hỏi: không phải ta nên bắt đầu đánh dấu từ bội số thứ hai của mỗi số nguyên tố sao?

Trả lời: Đây là phương án tối ưu khi sàng số nguyên tố. Dễ thấy, khi xóa các bội số của các số nguyên p , các bội số từ 2 đến $p - 1$ đều đã được

đánh dấu ở các lần trước đó. Ví dụ, ta không cần đánh dấu bội số thứ 3 của số nguyên tố 7, tức số 21, bởi vì số này đã được đánh dấu khi số nguyên tố 3 thực hiện việc đánh dấu các bội số.

Một cách để tối ưu hơn nữa thuật toán sàng chính là ta chỉ cần duyệt đến $\sqrt{n}$ với cách giải thích tương tự với phần trước.

```
for (int i = 2; i * i <= n; ++i) {
    if (notPrime[i] == 0)
        for (int j = i * i; j <= n; j += i)
            notPrime[j] = 1;
}
```

Mặc dù thuật toán sàng tối ưu hơn nhiều so với phương pháp thử sai, thuật toán lại sử dụng lượng bộ nhớ lớn, có độ phức tạp là $O(n)$, vì phải đánh dấu các số trong khoảng.

Thuật toán của Dijkstra

Thuật toán tìm số nguyên tố của Dijkstra là một thuật toán đầy "thanh lịch" trong việc tìm số nguyên tố.

Thuật toán áp dụng 2 tính chất:

- 2 là số nguyên tố đầu tiên và nhỏ nhất trong danh sách số nguyên tố
- Theo [định đề Bertrand](#) ¹⁸, với P_n là số nguyên tố thứ n , ta có nhận xét:

$$P_{n+1} < 2 \times P_n \leq P_n^2$$

Ta có mô tả thuật toán như sau:

Ta có một danh sách lưu các số nguyên tố dưới dạng cặp giá trị (P, tP) , với P là một số nguyên tố, và tP là bội số thứ t của số nguyên tố ấy ($t > 1$).

Ta lưu cặp số nguyên tố 2 và bội số thứ 2 của nó vào danh sách: $(2, 4)$.

Xét các số x từ 3 đến n :

- Nếu x nhỏ hơn bội số nhỏ nhất trong tất cả các bội số được lưu trong danh sách, x là một số nguyên tố. Khi này thêm cặp x và bội số thứ x của nó vào danh sách: (x, x^2) . Ví dụ: khi ta xét số 3, danh sách hiện tại lưu cặp giá trị $(2, 4)$. Vì $3 < 4$, 3 là số nguyên tố. Ta in 3 và thêm cặp giá trị $(3, 9)$ vào danh sách.
- Nếu x bằng một hoặc nhiều bội số nhất trong danh sách thì x không là một số nguyên tố. Khi này, tăng giá trị của (các) bội số một khoảng bằng số nguyên tố x tương ứng trong cặp giá trị. Ví dụ như khi ta xét tới số 4, danh sách của ta hiện tại đang lưu 2 cặp giá trị $(2, 4)$ và

¹⁷https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_l%C3%BD_s%E1%BB%91_nguy%C3%AA_n_t%E1%BB%91

¹⁸https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_l%C4%91%E1%BB%81_Bertrand



(3, 9). Vì 4 bằng bội số của 2 nên 4 không phải số nguyên tố. Khi đó cặp (2, 4) sẽ được cập nhật lại thành (2, 6).

Sau khi duyệt xong, các số trong danh sách sẽ lưu tất cả các số nguyên tố từ 1 đến n .

Ta có cài đặt trên C++, sử dụng `priority_queue`:

```
// lưu các cặp (tP, P), pq-min theo tP
priority_queue<
    pair<int, int>,
    vector<pair<int, int>>,
    greater<pair<int, int>>>
> pq;
pq.push({4, 2});

if (n <= 2) cout << 2 << '\n';

for (int i = 3; i <= n; ++i) {
    if (i < pq.top().first) {
        pq.push({i * i, i}); // Thêm (tP, P) vào danh sách
        cout << i << '\n';
    } else if (i == pq.top().first) {
        while (i == pq.top().first) {
            pair<int, int> a = pq.top();
            pq.pop();
            a.first += a.second; // thay đổi (tP, P) thành
                               // ((t + 1)P, P)
            pq.push(a);
        }
    }
}
```

Việc duyệt các số từ 1 đến n mất $O(n)$ thao tác, mỗi thao tác lại việc tìm bội số nhỏ nhất có độ phức tạp $O(\log n)$, nên ta có độ phức tạp $O(n \log n)$ cho thuật toán tìm số nguyên tố theo kiểu Dijkstra.

Thuật toán cũng phải lưu các số nguyên tố trong khoảng từ 1 đến n nên độ phức tạp bộ nhớ sẽ là $O(\pi(n)) \approx O(\frac{n}{\log_e n})$.

Ta tối ưu thuật toán bằng cách loại bỏ các số nguyên tố P khỏi danh sách nếu bội số hiện tại của nó tP lớn hơn n .

```
if (i < pq.top().F) {
    if (i * i <= n) pq.push({i * i, i});
    cout << i << '\n';
} else {
    while (pq.size() && i == pq.top().F) {
        auto a = pq.top();
        pq.pop();
        a.first += a.second;
        if (a.first < n) pq.push(a);
    }
}
```

So sánh với thuật toán Dijkstra

Thuật toán sàng số nguyên tố với thuật toán Dijkstra

So sánh với thuật toán sàng, thuật toán của Dijkstra sử dụng ít bộ nhớ hơn một cách đáng kể ($O(\pi(n)) \leq O(n)$). Đây là bởi thay vì phải lưu tất cả các bội của một số nguyên tố giống như thuật toán sàng, thuật toán của Dijkstra chỉ lưu một bội số cần thiết.

Tuy nhiên, vì cách lưu trữ và quản lý các bội số này khá phức tạp (phải sử dụng `priority_queue`), do đó độ phức tạp thời gian cũng lớn hơn so với thuật toán sàng.

Thử sai với thuật toán Dijkstra

Thuật toán của Dijkstra có độ phức tạp thời gian nhỏ hơn so với phương pháp thử sai. Tuy nhiên, phương pháp thử sai lại sử dụng ít bộ nhớ hơn so với thuật toán của Dijkstra.

Có thể thấy, thuật toán Dijkstra là một sự cân bằng giữa thời gian và bộ nhớ, với thuật toán chạy nhanh hơn phương pháp thử sai và sử dụng ít bộ nhớ hơn thuật toán sàng số nguyên tố.

Có nên sử dụng thuật toán tìm số nguyên tố của Dijkstra không?

Thuật toán của Dijkstra tuy "không có điểm yếu" nhưng chính vì sự cân bằng đó cũng khiến thuật toán "không có điểm mạnh" nếu so với hai thuật toán còn lại - hai thuật toán có điểm nổi trội rõ rệt thử sai sử dụng rất ít bộ nhớ, và thời gian chạy của sàng là khá nhỏ. Ngay cả trong bài viết của Dijkstra nói về thuật toán này cũng viết

"But everybody knows that the most efficient way to generate prime numbers is by using the Sieve of Eratosthenes."

(Tạm dịch: Ai cũng biết cách hiệu quả nhất để sinh số nguyên tố chính là sử dụng thuật toán sàng Eratosthenes.)

Tuy nhiên, đây cũng là một thuật toán rất hay ho mà bạn có thể học!

¹⁹<https://www.cs.utexas.edu/~EWD/ewd02xx/EWD249.PDF>



Tham khảo

[Notes on Structured Programming](#) ¹⁹ - Edsger
W. Dijkstra.



Tổng quan về cấu trúc dữ liệu

Trần Xuân Bách
University of Chicago - Class of 2028



Giới thiệu

Cấu trúc dữ liệu ²⁰ là một trong những kiến thức quan trọng nhất trong lập trình thi đấu bởi sự hữu dụng trong việc quản lí, truy cập và thay đổi dữ liệu một cách nhanh chóng. Tuy nhiên, sự phức tạp trong việc cài đặt chúng là sự bất tiện không hề nhỏ với tất cả mọi người. Do vậy, nhiều bạn thường sử dụng thư viện code (library) để rút ngắn thời gian cài đặt, nhưng một vấn đề mới xuất hiện là việc thiết kế một thư viện có đầy đủ tính năng, và áp dụng được với tất cả mọi tình huống.

Bài viết này sẽ tập trung về việc tổng quát hóa các cấu trúc dữ liệu thường gặp - **Sparse Table** ²¹, **Segment Tree** ²² - bằng lí thuyết nhóm (**Group theory** ²³), và cách cài đặt chúng.

Tác giả muốn gửi lời cảm ơn đến thư viện **Algorithms** ²⁴ của Aeren, nguồn cảm hứng cho bài viết này.

Ghi chú

Tuy không bắt buộc, nhưng độc giả nên đọc trước về các thuật toán và cấu trúc dữ liệu ở trên để đọc bài viết này một cách suôn sẻ nhất. Tác giả sẽ chỉ đề cập qua về ý tưởng chính của các thuật toán trên, mà không đi sâu vào lí do và cách hoạt động.

Để nhất quán với thư viện Algorithms trên, ta sẽ thống nhất rằng:

- Mảng đánh số bắt đầu từ 0 (0-index). Khi ta nói "Mảng A có N phần tử", ta mặc định rằng mảng được đánh số từ 0 đến $N - 1$.
- Mọi khoảng trong mảng đều là nửa khoảng $[l, r)$ (half-open interval). Cụ thể hơn, khoảng này chứa mọi số i thỏa mãn $l \leq i < r$.

Sparse Table

Bài toán cổ điển

Cho một mảng A có N số nguyên và Q truy vấn có dạng $[l, r)$. Với mỗi truy vấn, hãy tìm giá trị nhỏ nhất trong khoảng $[l, r)$.

Lời giải

Gọi $L = \lfloor \log N \rfloor + 1$. Ta định nghĩa mảng hai chiều ST có kích thước $[0, L) \times [0, N)$, với giá trị của $ST[j][i]$ là

$$\begin{aligned} ST[j][i] &= \min_{k \in [i, i+2^j)} A_k \\ &= \min \left(\min_{k \in [i, i+2^{j-1})} A_k, \min_{k \in [i+2^{j-1}, i+2^j)} A_k \right) \\ &= \min(ST[j-1][i], ST[j-1][i+2^{j-1}]) \end{aligned}$$

Để trả lời truy vấn $[l, r)$, gọi $j = \lfloor \log(r-l) \rfloor$, khi đó:

$$\begin{aligned} \min_{k \in [l, r)} A_k &= \min \left(\min_{k \in [l, l+2^j)} A_k, \min_{k \in [r-2^j, r)} A_k \right) \\ &= \min(ST[j][l], ST[j][r-2^j]) \end{aligned}$$

Độ phức tạp là $\mathcal{O}(N \log N + Q)$.

Cài đặt

```
struct sparse_table {
    int n;
    vector<vector<int>> st;

    void build(const vector<int> &a) {
        n = ssize(a);
        int l = __lg(n) + 1;
        st.assign({a});
        st.resize(1);
        for (auto j = 1; j < l; j++) {
            st[j].resize(n - (1 << j) + 1);
            for (auto i = 0; i + (1 << j) <= n; i++) {
                st[j][i] = min(
                    st[j-1][i], st[j-1][i + (1 << (j-1))]
                );
            }
        }
    }

    int query(int l, int r) {
        int j = __lg(r-l);
        return min(st[j][l], st[j][r - (1 << j)]);
    }
};
```

Bài toán tổng quát

Đầu tiên, ta phải làm tổng quát bài toán trước. Ta biết rằng mảng A có thể chứa nhiều kiểu biến/dữ liệu khác nhau (số nguyên, số hữu tỉ, số thực, số phức, ...), và truy vấn có thể hỏi về các hàm khác nhau (min, max, gcd, ...). Gọi kiểu biến trên là T và hàm là merge, ta có bài toán tổng quát như sau:

²⁰<https://wiki.vnoi.info/algo/data-structures/data-structures-overview>

²¹<https://wiki.vnoi.info/algo/data-structures/rmq>

²²<https://wiki.vnoi.info/algo/data-structures/segment-tree-basic>

²³https://en.wikipedia.org/wiki/Group_theory

²⁴<https://github.com/Aeren1564/Algorithms/>



Cho một mảng A có N phần tử thuộc T ($A_i \in T$) và Q truy vấn có dạng $[l, r]$. Với mỗi truy vấn, hãy tìm $\text{merge}(A_{[l,r]}) = \text{merge}(A_l, \dots, A_{r-1})$.

Nhận xét

Ta có một vài nhận xét như sau:

- merge là một phép toán hai ngôi ([binary operation](#)²⁵) trên T ; nói cách khác,

$$\text{merge} : T \times T \rightarrow T$$

- Để định nghĩa $\text{merge}(A_{[l,r]})$, ta cần tính chất giao hoán ([associative](#)²⁶) của merge :

$$\text{merge}(a, \text{merge}(b, c)) = \text{merge}(\text{merge}(a, b), c)$$

$$\forall a, b, c \in T$$

Tính chất này là cần thiết để biểu thức trên có cách biểu diễn là $\text{merge}(a, b, c)$; và mở rộng hơn là $\text{merge}(a_1, \dots, a_n) = \text{merge}(a_1, \text{merge}(\dots, \text{merge}(a_{n-1}, a_n)))$.

- Để thuận tiện, ta định nghĩa phần tử đơn vị của T , gọi là T_{id} . Phần tử này thỏa mãn:

$$\text{merge}(a, T_{\text{id}}) = \text{merge}(T_{\text{id}}, a) = a$$

$$\forall a \in T$$

Từ đây ta có định nghĩa rằng $T(A_{[i,i]}) = T_{\text{id}}$

Ba nhận xét trên chính là ba tiên đề của [monoid](#)²⁷. Nói cách khác, T là một monoid với phép toán merge và phần tử đơn vị T_{id} .

Ngoài ra, để Sparse Table vẫn hoạt động chính xác, ta phải có thêm điều kiện sau:

- Xét $A = [T_{\text{id}}, a, T_{\text{id}}]$ ($a \in T$) và truy vấn $[l, r] = [0, 3]$. Theo thuật toán Sparse Table, ta có: $\text{merge}(A_{[0,3]})$
 $= \text{merge}(\text{merge}(A_{[0,2]}), \text{merge}(A_{[1,3]}))$
 $= \text{merge}(T_{\text{id}}, a, T_{\text{id}})$
 $= \text{merge}(\text{merge}(T_{\text{id}}, a), \text{merge}(a, T_{\text{id}}))$

$$a = \text{merge}(a, a)$$

Vậy merge phải thỏa mãn tính chất [idempotence](#)²⁸: $a = \text{merge}(a, a) \forall a \in T$.

Điều kiện này cũng là điều kiện đủ để bài toán có thể được giải bằng Sparse Table. Thật vậy, ta có:

$$\begin{aligned} & \text{merge}(\text{ST}[j][l], \text{ST}[j][r - 2^j]) \\ &= \text{merge}(A_{[l, l+2^j]}, A_{[r-2^j, r]}) \\ &= \text{merge}(\text{merge}(A_{[l, r-2^j]}, A_{[r-2^j, l+2^j]}), \\ & \quad \text{merge}(A_{[r-2^j, l+2^j]}, A_{[l+2^j, r]})) \\ &= \text{merge}(A_{[l, r-2^j]}, \text{merge}(A_{[r-2^j, l+2^j]}, A_{[r-2^j, l+2^j]}), A_{[l+2^j, r]}) \\ &= \text{merge}(A_{[l, r-2^j]}, A_{[r-2^j, l+2^j]}, A_{[l+2^j, r]}) \\ &= \text{merge}(A_{[l, r]}) \end{aligned}$$

Một số ví dụ của bài toán thỏa mãn những điều kiện trên là:

- T là số thực, merge là $\min$, T_{id} là ∞ .
- T là số thực, merge là $\max$, T_{id} là $-\infty$.
- T là số tự nhiên, merge là $\gcd$, T_{id} là 0.
- T là bitmask có D bit, merge là AND, T_{id} là $2^D - 1$.
- T là bitmask có D bit, merge là OR, T_{id} là 0.

Cài đặt

[Bạn đọc có thể tham khảo thêm code tại đây](#)²⁹

```
// TT là loại của merge: T x T -> T
template <class T, class TT> struct sparse_table {
    int n;
    vector<vector<T>> st;
    TT merge; // Binary operation
    T T_id; // Identity element

    sparse_table(TT merge, T T_id)
        : merge(merge)
        , T_id(T_id) {}

    void build(const vector<T> &a) {
        n = ssize(a);
        int l = __lg(n) + 1;
        st.assign({a});
        st.resize(l);
        for (auto j = 1; j < l; j++) {
            st[j].resize(n - (1 << j) + 1);
            for (auto i = 0; i + (1 << j) <= n; i++) {
                st[j][i] = merge(
                    st[j - 1][i], st[j - 1][i + (1 << (j - 1))]
                );
            }
        }
    }

    T query(int l, int r) {
        if (l == r) return T_id;
        int j = __lg(r - l);
        return merge(st[j][l], st[j][r - (1 << j)]);
    }
};
```

²⁵https://en.wikipedia.org/wiki/Binary_operation

²⁶https://en.wikipedia.org/wiki/Associative_property

²⁷<https://en.wikipedia.org/wiki/Monoid>

²⁸<https://en.wikipedia.org/wiki/Idempotence>

²⁹https://github.com/Aeren1564/Algorithms/blob/master/Algorithm_Implementations_Cpp/Data_Structure/Sparse_Table/sparse_table.cpp



Không có nhiều thay đổi giữa cách cài đặt tổng quát này so với ban đầu, nhưng ta có thể dùng nó cho nhiều thể loại Sparse Table khác nhau:

```
// int array, minimum query
auto st_int_min = sparse_table(
    /* merge */ [](
        int t1, int t2
    ) { return t1 < t2 ? t1 : t2; },
    /* T_id */ numeric_limits<int>::max()
);

// long double array, maximum query
auto st_ld_max = sparse_table(
    /* merge */ [](
        long double t1, long double t2
    ) { return t1 > t2 ? t1 : t2; },
    /* T_id */ numeric_limits<long double>::min()
);

// long long array, gcd query
auto st_ll_gcd = sparse_table(
    /* merge */ [](
        long long t1, long long t2
    ) { return gcd(t1, t2); },
    /* T_id */ 0ll
);

// int array, and query
auto st_int_and = sparse_table(
    /* merge */ [](int t1, int t2) { return t1 & t2; },
    /* T_id */ ~0
);

// int array, or query
auto st_int_or = sparse_table(
    /* merge */ [](int t1, int t2) { return t1 | t2; },
    /* T_id */ 0
);
```

Segment Tree

Bài toán cổ điển

Ta xét một trong những bài toán cơ bản nhất thường gặp khi bắt đầu học Segment Tree: Point update, range sum.

Cho một mảng A có N số nguyên và Q truy vấn thuộc 1 trong 2 dạng:

- 1 i x : Gán $A_i := x$.
- 2 u v : Tính giá trị lớn nhất của mảng A trong khoảng $[u, v)$.

Lời giải

Ta sẽ xây một cây nhị phân đầy đủ (full binary tree³⁰), mỗi đỉnh lưu tổng của một đoạn con của A như sau:

1. Đỉnh gốc của cây là 1, quản lý đoạn $[l_1, r_1) = [0, N)$.

2. Nếu đỉnh u quản lý đoạn $[l_u, r_u)$, và $r_u - l_u > 1$, thì u có 2 con. Định nghĩa $m_u = \lfloor \frac{l_u + r_u}{2} \rfloor$,

hai con của đỉnh u là:

- Con trái là đỉnh $2u$, quản lý đoạn $[l_u, m_u)$.
- Con phải là đỉnh $2u + 1$, quản lý đoạn $[m_u, r_u)$.

3. Nếu đỉnh u quản lý đoạn $[i, i + 1)$, thì u là lá của cây.

Cây này thỏa mãn những tính chất sau:

- Độ sâu của cây là $\mathcal{O}(\log N)$.
- Với mọi đoạn con $[l, r)$, ta có thể tách nó ra thành $\mathcal{O}(\log N)$ đoạn được quản lý bởi các đỉnh trên cây.

Do vậy, ở mỗi đỉnh u , ta sẽ lưu

$$\begin{aligned} \text{data}_u &= \max(A_{[l_u, r_u)}) \\ &= \max(\max(A_{[l_u, m_u)}), \max(A_{[m_u, r_u)})) \\ &= \max(\text{data}_{2u}, \text{data}_{2u+1}) \end{aligned}$$

Với mỗi truy vấn 1, ta chỉ cần cập nhật data của $\mathcal{O}(\log N)$ đỉnh trên đường đi từ gốc đến đỉnh $[i, i + 1)$. Với truy vấn 2, ta sẽ tính giá trị lớn nhất của $\mathcal{O}(\log N)$ đỉnh bao phủ đoạn $[u, v)$.

Độ phức tạp là $\mathcal{O}(N \log N + Q \log N)$.

Cài đặt

```
struct segment_tree {
    int n;
    vector<int> data;

    void build(const vector<int> &a) {
        n = ssize(a);
        data.resize(1 << __lg(4 * n - 1));
        auto recurse =
            [&](auto &&self, int id, int l, int r) -> void {
                if (r - l == 1) {
                    data[id] = a[l];
                    return;
                }
                int mid = (l + r) / 2;
                self(self, id * 2, l, mid);
                self(self, id * 2 + 1, mid, r);
                data[id] = max(data[id * 2], data[id * 2 + 1]);
            };
        recurse(recurse, 1, 0, n);
    }

    void update(int i, int x) {
        auto recurse =
            [&](auto &&self, int id, int l, int r) -> void {
                if (r - l == 1) {
                    data[id] = x;
                    return;
                }
                int mid = (l + r) / 2;
                if (i < mid) self(self, id * 2, l, mid);
                else
                    self(self, id * 2 + 1, mid, r);
                data[id] = max(data[id * 2], data[id * 2 + 1]);
            };
        recurse(recurse, 1, 0, n);
    }
};
```

³⁰https://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees



```
}

int query(int u, int v) {
    auto recurse =
        [&](auto &&self, int id, int l, int r) -> int {
            if (v <= l or r <= u)
                return numeric_limits<int>::min();
            if (u <= l and r <= v) return data[id];
            int mid = (l + r) / 2;
            return max(
                self(self, id * 2, l, mid),
                self(self, id * 2 + 1, mid, r)
            );
        };
    return recurse(recurse, 1, 0, n);
};
```

Bài toán tổng quát

Ta có bài toán tổng quát như sau:

Cho một mảng A có N phần tử thuộc T và Q truy vấn thuộc 1 trong 2 dạng:

- 1 i x : Gán $A_i := x$.
- 2 u v : Tính $\text{merge}(A_{[u,v]})$.

Nhận xét

Tương tự như phần Sparse Table, ta thấy T phải là một monoid với phép toán merge và phần tử đơn vị T_{id} .

Cài đặt

Việc này tương tự với việc chúng ta đã làm ở phần Sparse Table. Để cho ngắn gọn, ta chỉ xét phần cài đặt cho truy vấn 2:

```
template <class T, class TT> struct segment_tree {
    ... TT merge;
    T T_id;
    segment_tree(TT merge, T T_id)
        : merge(merge)
        , T_id(T_id) {} ... T query(int u, int v) {
        auto recurse =
            [&](auto &&self, int id, int l, int r) -> T {
                if (v <= l or r <= u) return T_id;
                if (u <= l and r <= v) return data[id];
                int mid = (l + r) / 2;
                return merge(
                    self(self, id * 2, l, mid),
                    self(self, id * 2 + 1, mid, r)
                );
            };
        return recurse(recurse, 1, 0, n);
    }
};
```

Dưới đây là một số dạng Segment Tree thường gặp nhất:

```
// max query
auto seg_max = segment_tree(
    /* merge */ [](
        int t1, int t2
    ) { return t1 > t2 ? t1 : t2; },
```

```
/* T_id */ numeric_limits<int>::min()
);

// max query, và tìm vị trí của max. T là một cặp số
// lưu (max, vị trí max).
auto seg_maxpos = segment_tree {
    /* merge */ [](
        const pair<int, int> &t1,
        const pair<int, int> &t2
    ) { return t1 > t2 ? t1 : t2; },
    /* T_id */ pair{numeric_limits<int>::min(), -1};
};

// max subarray query. T là một struct lưu {max
// subarray, max prefix, max suffix, sum}.
struct T {
    int max_subarray, max_prefix, max_suffix, sum;
};

auto seg_maxsubarray = segment_tree{
    /* merge */ [](const T &t1, const T &t2) {
        return T{
            /* max_subarray */ max(
                {t1.max_subarray,
                 t2.max_subarray,
                 t1.max_suffix + t2.max_prefix}
            ),
            /* max_prefix */
            max({t1.max_prefix, t1.sum + t2.max_prefix}),
            /* max_suffix */
            max({t1.max_suffix + t2.sum, t2.max_suffix}),
            /* sum */ t1.sum + t2.sum
        };
    },
    /* T_id */ T{
        /* max_subarray */ numeric_limits<int>::min(),
        /* max_prefix */ numeric_limits<int>::min(),
        /* max_suffix */ numeric_limits<int>::min(),
        /* sum */ 0
    }
};
```

Lazy Segment Tree

Segment Tree cơ bản chỉ giải quyết được những bài toán có truy vấn cập nhật một phần tử duy nhất. Tuy nhiên, nhiều bài toán - như ví dụ sau đây - phải cần dùng đến Lazy Segment Tree:

Bài toán cổ điển

Cho một mảng A có N số nguyên và Q truy vấn thuộc 1 trong 2 dạng:

- 1 u v x : Với mọi $i \in [u, v)$, tăng giá trị của A_i lên x .
- 2 u v : Tính giá trị lớn nhất của mảng A trong khoảng $[u, v)$.



Lời giải

Khi thực hiện truy vấn 1, để giữ độ phức tạp không quá $\mathcal{O}(\log N)$, việc đệ quy phải dừng lại khi ta gặp đỉnh $[l, r]$ nằm hoàn toàn trong $[u, v]$. Do vậy, ở mỗi đỉnh, ta sẽ lưu thêm một biến lazy để lưu lại những cập nhật mà ta chưa "đẩy" từ đỉnh này xuống các đỉnh con của nó. Trước khi ta truy cập vào data của một đỉnh u bất kì, ta phải "đẩy" các cập nhật được lưu trong lazy của cha nó xuống data của u , và cả lazy của u nữa (để đẩy xuống các con của u).

Bạn đọc có thể đọc chi tiết hơn tại [đây](#)³¹

Cài đặt

```
struct segment_tree_lazy {
    int n;
    vector<int> data;
    vector<int> lazy;
    ...
    // Đẩy các cập nhật đang chờ ở đỉnh id xuống con
    void push(int id) {
        data[id * 2] += lazy[id];
        data[id * 2 + 1] += lazy[id];
        lazy[id * 2] += lazy[id];
        lazy[id * 2 + 1] += lazy[id];
        lazy[id] = 0;
    }

    void update(int u, int v, int x) {
        auto recurse =
        [&](auto &&self, int id, int l, int r) -> void {
            if (v <= l or r <= u) return;
            if (u <= l and r <= v) {
                data[id] += x;
                lazy[id] += x;
                return;
            }
            push(id);
            int mid = (l + r) / 2;
            self(self, id * 2, l, mid);
            self(self, id * 2 + 1, mid, r);
            data[id] = max(data[id * 2], data[id * 2 + 1]);
        };
        recurse(recurse, 1, 0, n);
    }
    ...
};
```

Bài toán tổng quát

Nhận xét rằng tổng quát của truy vấn 1 chỉ là ta thực hiện một hàm cập nhật nào đó lên một đoạn con $A_{[u,v]}$. Gọi hàm đó là change , ta có bài toán sau:

Cho một mảng A có N phần tử thuộc T và Q truy vấn thuộc 1 trong 2 dạng:

- 1 $u\ v\ \text{change}$: Với mọi $i \in [u, v]$, gán $A_i := \text{change}(A_i)$.
- 2 $u\ v$: Tính $\text{merge}(A_{[u,v]})$.

Nhận xét

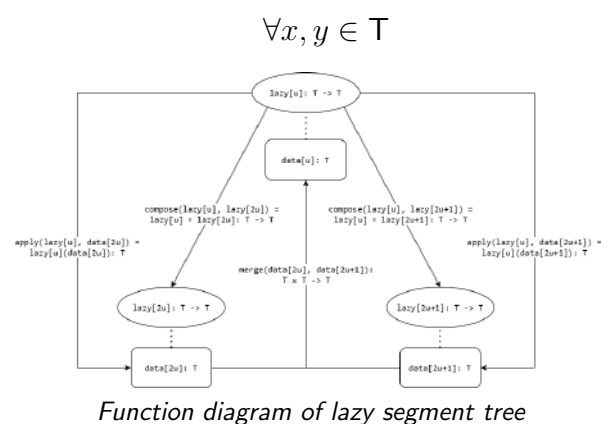
- change là một [transformation/self-map](#)³² trên T . Nói cách khác, $\text{change} : T \rightarrow T$. Gọi tập hợp các hàm này là U . Để ý rằng lazy cũng là một hàm $T \rightarrow T$ trong U .
- Khi đẩy lazy của cha xuống data của đỉnh con, ta chỉ cần tính giá trị của hàm lazy với data đó. Khi đẩy lazy của cha xuống lazy của con, ta cần tìm một hàm mới tương đương với việc ta thực hiện hàm lazy của đỉnh con trước, sau đó thực hiện hàm lazy của đỉnh cha lên kết quả trung gian này. Đây chính là phép hàm hợp ([function composition](#)³³), với kết quả là một hàm thuộc U .
- Sau khi ta đẩy lazy xong, lazy của đỉnh cha phải được gán với hàm đồng nhất ([identity function](#)³⁴). Gọi hàm này là U_{id} .

Thử thách nhỏ cho bạn đọc: Dựa trên 3 nhận xét trên, U phải là cấu trúc đại số gì?

Ngoài ra, để Lazy Segment Tree hoạt động, ta phải có một điều kiện quan trọng sau:

- Việc thực hiện hàm lazy trên tổng merge của một đoạn con của A tương đương với việc lấy tổng merge của từng phần tử một, sau khi đã cập nhật lazy riêng. Nói cách khác:

$$\text{lazy}(\text{merge}(x, y)) = \text{merge}(\text{lazy}(x), \text{lazy}(y))$$



Cài đặt

Bạn đọc có thể tham khảo thêm tại [đây](#)³⁵.

³¹<https://wiki.vnoi.info/algo/data-structures/segment-tree-extend#ph%C3%A2n-t%C3%ADch>

³²[https://en.wikipedia.org/wiki/Transformation_\(function\)](https://en.wikipedia.org/wiki/Transformation_(function))

³³https://en.wikipedia.org/wiki/Function_composition

³⁴https://en.wikipedia.org/wiki/Identity_function

³⁵https://github.com/Aeren1564/Algorithms/blob/master/Algorithm_Implementations_Cpp/Data_Structure/Segment_Tree/snippet



```
template <
class T,
class U,
class TT,
class UT,
class UU>
struct lazy_segment_tree {
    int n;
    vector<T> data;
    vector<U> lazy;
    TT merge;
    T T_id;
    UT apply;
    UU compose;
    U U_id; // identity function

    lazy_segment_tree(
        TT merge, T T_id, UT apply, UU compose, U U_id
    )
        : merge(merge)
        , T_id(T_id)
        , apply(apply)
        , compose(compose)
        , U_id(U_id) {}

    void build(const vector<T> &a) {
        n = ssize(a);
        data.assign(1 << __lg(4 * n - 1), T_id);
        lazy.assign(1 << __lg(4 * n - 1), U_id);
        auto recurse =
            [&](auto &&self, int id, int l, int r) -> void {
                if (r - l == 1) {
                    data[id] = a[l];
                    return;
                }
                int mid = (l + r) / 2;
                self(self, id * 2, l, mid);
                self(self, id * 2 + 1, mid, r);
                data[id] = merge(data[id * 2], data[id * 2 + 1]);
            };
        recurse(recurse, 1, 0, n);
    }

    void push(int id) {
        data[id * 2] = apply(lazy[id], data[id * 2]);
        data[id * 2 + 1] =
            apply(lazy[id], data[id * 2 + 1]);
        lazy[id * 2] = compose(lazy[id], lazy[id * 2]);
        lazy[id * 2 + 1] =
            compose(lazy[id], lazy[id * 2 + 1]);
        lazy[id] = U_id;
    }

    void update(int u, int v, U change) {
        assert(0 <= u and u <= v and v <= n);
        if (u == v) return;
        auto recurse =
            [&](auto &&self, int id, int l, int r) -> void {
                if (v <= l or r <= u) return;
                if (u <= l and r <= v) {
                    data[id] = apply(change, data[id]);
                    lazy[id] = compose(change, lazy[id]);
                    return;
                }
                push(id);
                int mid = (l + r) / 2;
                self(self, id * 2, l, mid);
                self(self, id * 2 + 1, mid, r);
                data[id] = merge(data[id * 2], data[id * 2 + 1]);
            };
        recurse(recurse, 1, 0, n);
    }

    T query(int u, int v) {
        assert(0 <= u and u <= v and v <= n);
        if (u == v) return T_id;
        auto recurse =
            [&](auto &&self, int id, int l, int r) -> int {
                if (v <= l or r <= u) return T_id;
```

```
                if (u <= l and r <= v) return data[id];
                push(id);
                int mid = (l + r) / 2;
                return merge(
                    self(self, id * 2, l, mid),
                    self(self, id * 2 + 1, mid, r)
                );
            };
        return recurse(recurse, 1, 0, n);
    }
};
```

Ta có thể cài đặt nhiều dạng Lazy Segment Tree như sau:

```
// range add, range max (bài toán cổ điển ở trên)
auto seg_add_max = lazy_segment_tree(
    /* merge */ [](
        int t1, int t2
    ) { return t1 > t2 ? t1 : t2; },
    /* T_id */ numeric_limits<int>::min(),
    /* apply */ [](int u, int t) { return u + t; },
    /* compose */ [](int u1, int u2) { return u1 + u2; },
    /* U_id */ 0
);

// range add, range sum. T là một cặp số lưu (tổng, độ dài đoạn)
auto seg_add_sum = lazy_segment_tree(
    /* merge */
    [](
        const pair<int, int> &t1, const pair<int, int> &t2
    ) {
        return pair{
            t1.first + t2.first, t1.second + t2.second
        };
    },
    /* T_id */ pair{0, 0},
    /* apply */
    [](int u, const pair<int, int> &t) {
        return pair{t.first + u * t.second, t.second};
    },
    /* compose */ [](int u1, int u2) { return u1 + u2; },
    /* U_id */ 0
);
```



Phỏng vấn Nguyễn Khánh Phúc

Interviewer: Nguyễn Trung Quân – Ban biên tập Tạp chí VNOI



Trước tiên thì em có thể giới thiệu đôi lời về bản thân, cũng như những thành tích nổi bật trong lập trình thi đấu không?

Em xin tự giới thiệu, em tên là Nguyễn Khánh Phúc. Hiện tại, em đang học lớp 9D trường THCS Trương Hán Siêu. Về thành tích của em khi theo đuổi lập trình thi đấu, em đã giành chức vô địch Olympic miền Trung Tây Nguyên bảng chuyên Tin năm 2024. Ngoài ra, em đã được tham dự chung kết VNOI Cup tháng 8 vừa rồi tại TP. Hồ Chí Minh. Ngoài ra, em cũng có giải nhất của hội thi Tin Học Trẻ toàn quốc bảng B năm 2024.

Được biết rằng em là một trong những thí sinh nhỏ tuổi nhất của VNOI Cup năm nay. Em đã bắt đầu học tin từ bao giờ?

Em bắt đầu nghiên cứu tin học từ năm lớp ba. Hồi đấy, em là một học sinh có thành tích trong các kỳ thi giải toán trên mạng. Em tham dự kỳ thi Violympic cấp Quốc gia lớp 4, đạt được huy chương bạc. Sau đấy, em được các thầy cô của bộ môn tin học chú ý, gọi vào để học về lập trình Scratch. Với lập trình Scratch, em giành được hai giải nhất bảng A Tin Học Trẻ vào năm 2019 và 2021. Nhờ những thành tích đó, em được các thầy cô trong tổ tin học nhà trường chú ý, động viên vào học cùng đội tuyển HSG lớp 9. Từ đó, em bắt đầu nghiên cứu về lập trình thi đấu.

Khi bắt đầu lập trình, em được làm quen với ngôn ngữ Scratch. Từ Scratch, em chuyển qua C++ có những khó khăn nào không?

Em thấy rằng việc chuyển từ Scratch sang C++ có lợi thế hơn so với việc học C++ ngay từ đầu, vì khi em học Scratch, em đã có tư duy về thuật toán. Vào năm lớp 5, em đã 'tự chế' tìm kiếm nhị phân và áp dụng trên Scratch mà không cần sự trợ giúp. Khi có tư duy, chuyển sang một ngôn ngữ lập trình mới sẽ không quá khó khăn. Khi tiếp xúc với C++ - nổi tiếng là ngôn ngữ lập trình khó, cũng như những bạn khác, em thấy rắc rối trong việc quản lý các mảng, biến. Học cú pháp, tư duy của C++ khá khó khăn trong những tháng đầu tiên.

Anh thấy rằng em gặp khó vì syntax (cú pháp) của C++. Em mất bao lâu để làm quen với C++?

Vào khoảng tháng 2/2022, em bắt đầu code 'Hello World!', và hai tháng sau, em có đủ khả năng làm toàn bộ đề HSG lớp 9 tỉnh em (Ninh Bình). Do em không được học C++ một cách bài bản mà tự học hoàn toàn, nên về mặt cú pháp thì em vẫn còn thiếu nhiều. Các anh chị ngày nay khi sắp xếp mảng có điều kiện tùy chỉnh thì toàn dùng lambda để viết hàm so sánh cho hàm sort, nhưng em vẫn dùng phương pháp truyền thông 'bool cmp' ạ!

Phương pháp Lambda khá phổ biến, nếu suy nghĩ nhanh thì code Lambda sẽ rất tiện! Tuy vậy, Lambda liên quan đến tính năng của C++, không có liên kết quá nhiều với thuật toán.





Em bắt đầu lập trình từ rất sớm. Nhưng những bài toán trong đề Học sinh giỏi có thể không phù hợp với khả năng hồi lớp 6, 7 của em; vì các phép tính toán cần thiết chưa có mặt trong chương trình THCS. Điều đó có gây khó khăn gì cho em hay không?



Đối với phiên bản lớp 6 của em, các bài tập đó sẽ gây khó khăn một chút. Hồi đấy, em vẫn còn hơi yếu phần đồng dư thức (modulo) nên đoạn tháng 6/2022, em bắt đầu học nghịch đảo Modulo và hình học tính toán; hai phần đó đã làm khó em rất nhiều vì em chưa đủ kiến thức toán. Em giải quyết vấn đề bằng cách vẫn theo học Đội tuyển Toán của trường. Hiện tại đã là năm thứ tư, em cũng đã có một số thành tích nhất định trong cuộc thi Học sinh giỏi Toán thành phố. Nhờ ba năm học bồi dưỡng, em đã có thể làm được những bài 'nặng toán' không có trong chương trình THCS.

Trong tính toán độ phức tạp, có một số hàm logarit, căn bậc 2, ... thì chỉ được học năm lớp 9 thôi. Khi tiếp cận với Lập trình thi đấu, em đã biết các khái niệm đó từ trước chưa?

Hồi lớp 5, em rất thích toán. Em được mẹ em nói về logarit và căn bậc 2. Tuy vậy thì em chỉ biết ở mức kiến thức nền, khả năng giải toán vẫn chưa đạt tới HSG lớp 9 hoặc 10 được.

Như đã chia sẻ, em làm quen với các khái niệm tin học trước khi bồi đắp những kiến thức toán cần thiết. Khi làm bài kiểm tra trên lớp, thi học kì, ... , em có

cảm thấy được sự khác biệt hay không?

Toán học cho Tin học và Toán học trên lớp cực kì khác biệt. Có những phần mà Tin học yêu cầu rất nhiều mà chương trình Toán trên lớp không chú trọng, và ngược lại. Tuy nhiên, quay lại với các bài toán ở chương trình đại trà sẽ không quá khó khăn với các bạn theo đuổi lập trình thi đấu. Khi bạn đã có khả năng lập trình thi đấu, việc làm toán chương trình đại trà không phải là một vấn đề quá lớn. Em không phải là dân chuyên Toán, không học Toán nhiều. Dù vậy, em vẫn thử sức và đạt kết quả tốt trong kì thi HSG Toán thành phố. Các bạn sẽ thoải mái vượt qua thôi.

Có nhiều bạn học sinh chuyên Toán chuyển sang học Tin và bứt lên rất nhanh dù có khởi đầu chậm. Theo em, học toán tới mức độ nào sẽ “đủ” để chinh phục môn tin?

Như em đã nói ở trước, toán trong tin rất khác toán trong các kì thi HSG và Olympic. Do vậy, các yếu tố còn phải phụ thuộc vào trình độ từng mảng của mỗi người đến đâu. Em tự thấy mình khá mạnh phần tổ hợp - em không quá khó khăn trong việc làm các đề, kể cả HSG cấp 3. Ngược lại, bắt đầu thức làm em chật vật ngay từ các đề HSG Thành phố lớp 9. Tóm lại, đối với em, yêu cầu đối với phần tổ hợp và số học sẽ được chú trọng nhiều nhất. Các phần khác sẽ không đóng góp nhiều đến thành tích lập trình thi đấu đâu.

Trong quá trình theo đuổi Lập trình thi đấu, nguồn học tập chủ yếu của em ở đâu?

Những ngày đầu tiên, em lấy đề HSG lớp 9 của các tỉnh (tất nhiên là không phải Hà Nội à!), sau đó tập nghĩ và luyện code chắc tay hơn. Sau khoảng bốn tháng học, em được người thân giới thiệu cuốn



Giải thuật và Lập trình của thầy Lê Minh Hoàng, khi em đã đạt được trình độ cao hơn. Đến ngày nay, em vẫn chưa đọc hết quyển đấy vì nó khá khó, đặc biệt là thuật toán tìm cặp ghép Blossom. Tháng 5/2022, em bắt đầu luyện trên VNOJ và tham gia contest đầu tiên trên đấy. Ngoài ra, em cũng chuyển sang làm Codeforces³⁶ với nguồn bài tập đa dạng hơn. Hiện tại thì em làm trên các Online Judge Codeforces và Atcoder.

Trong quá trình tự học, em có đọc các bài viết trên VNOI Wiki không?

Em có đọc. Với các thuật toán khó, VNOI Wiki là nguồn tài liệu tiếng Việt duy nhất thật sự hiệu quả. Cách duy nhất để có thể tiếp cận đến những kỹ thuật cao như Heavy-light Decomposition (Phân tách cây nặng nhẹ - HLD) hay Centroid Decomposition (Phân tách trọng tâm) là các blog trên Codeforces, CP³⁷ - Algorithm hoặc USACO. Em không phải là người có lợi thế về tiếng Anh. Đọc những bài tiếng Anh luôn khó khăn hơn các bài tiếng Việt. Nên khi VNOI Wiki bắt đầu hoàn thiện, em chuyển qua đọc trên ấy nhiều hơn. Tuy vậy, VNOI Wiki vẫn còn một số phần chưa đầy đủ. Luồng cực đại chi phí cực tiểu (Min-cost Max Flow), FFT hay XOR Basis không được viết nhiều nhưng VNOI Wiki sẽ đáp ứng đầy đủ các kỹ thuật không quá phức tạp cho em.

Em có xem việc tự học là trở ngại hay thuận lợi trên con đường lập trình thi đấu?

Em tin rằng để đạt được mức cao hơn trong lập trình thi đấu, tất cả mọi người đều phải có tinh thần tự học. Dù đang tập trung ôn thi cuối cấp và thi vào chuyên, em luôn dành 4 tiếng mỗi ngày cho lập trình thi đấu. Tuy nhiên, việc phải tự học là một trở ngại khá lớn. Mò mẫm các thuật toán, học không theo sự chỉ dạy của bất cứ ai, bài này cần suy nghĩ theo hướng nào, ... sẽ chậm hơn rất nhiều so với việc có người giúp đỡ.

VNOI mới cho ra mắt VNOI Roadmap, bao gồm lộ trình từ cơ bản đến nâng cao, có cả những thuật toán cho vòng 2! Em đã sử dụng VNOI Roadmap lần nào chưa, và cảm nhận của em như thế nào?

VNOI Roadmap là một dự án vô cùng hữu ích cho các bạn ở mọi trình độ. Em không may mắn lắm khi bắt đầu vài năm trước khi VNOI Cup tồn tại (cười)! Thuật toán trên VNOI Roadmap khá đầy đủ, bao gồm cả thuật toán em đã học và chưa học. Tuy vậy, VNOI Roadmap chỉ liệt kê lộ trình học các thuật toán cần thiết, còn để có kỹ năng, tư duy làm bài thì cách duy nhất là phải làm thật nhiều bài.

Em có muốn trở thành TNV VNOI để đóng góp cho cộng đồng Tin học Việt Nam không?

Nếu có cơ hội, em muốn thử sức với vai trò TNV VNOI. Em cảm thấy VNOI là một dự án vô cùng nhân văn, khơi dậy niềm đam mê Tin học cho rất nhiều bạn trẻ trên mọi miền tổ quốc, điển hình là em. VNOJ là nguồn đề đầu tiên em làm. Em muốn sau này có thể đóng góp lại cho VNOI, cũng như giúp đỡ các bạn trẻ bắt đầu môn Tin học, như cái cách em được giúp đỡ từ thuở ban đầu.



Em có hai năm tham gia VNOI CUP – 2023 và 2024. Năm 2023, em dành được top 75 trong hai contests đầu tiên. VNOI Cup 2024, em bứt phá rất nhiều, dành top 7 và 24 ở hai vòng loại năm nay. Giữa hai mùa VNOI Cup, em đã có sự thay đổi nào trong cách học, luyện tập và làm bài hay không?

Sau VNOI Cup 2023, em bắt đầu tập trung luyện bài nhiều hơn; ngoài ra, em lựa chọn những bài mà thật sự đẩy trình độ của em lên. Trước mùa VNOI Cup năm đó, em luyện bài trên Codeforces nhiều,

³⁶Codeforces: một trang web tổ chức các cuộc thi lập trình thi đấu với các dạng đề, bài tập đa dạng.

³⁷CP: Competitive Programming - Lập trình thi đấu



nhưng chỉ ở mức 1400 tới 1500 rating (!). Tự em cũng cảm thấy các bài đó dễ so với trình độ của em. Sau VNOI Cup 2023 và mùa Tin Học Trẻ năm 2023 với kết quả không được khả quan, em thay đổi chiến thuật luyện tập của mình. Những ngày đầu tiên, em làm bài ở mức 1900 tới 2000, sau đó đẩy độ khó dần. Trong khoảng 6-8 tháng, em đã giải được gần 400 bài. Rating em có thể giải ổn định ở mức là 2400.

Em hay chọn bài để luyện một cách ngẫu nhiên, không quan tâm đến tag; hay luyện tập theo chủ đề - hôm nay học Quy hoạch động, hôm sau học đồ thị?



Em là người học theo cách đầu tiên. Trong lập trình thi đấu, các bài 2600 Codeforces đổ xuống không yêu cầu kĩ thuật, các thuật toán “dị”. Em chỉ gặp một số bài Xor Basis, Luồng, FFT, ... Độ khó của bài nằm ở việc tìm-ra-thuật-toán-cần-dùng. Ví dụ, em gặp một bài DFS, nhưng đọc vào lại chả giống DFS một chút nào. Nếu dùng tag, có thể em sẽ nghĩ ra ngay lập tức. Nhưng trong kì thi thật, em sẽ không được dùng tag mà phải tự tìm ra đường đi cho mình.

Gu làm bài của em phụ thuộc vào rating và mang tính random rất cao, không theo một dạng nhất định.

Vậy em đã bao giờ thử làm những contest Educational trên Atcoder hay cả VNOJ? Em cảm thấy các bài đó như thế nào?

Em rất ít khi làm các Educational Contest liên quan đến một tag vì các thuật xuất hiện trong đó em đã biết rồi (!). Do vậy, em hiếm khi làm các kì thi này. Em nhớ một trường hợp ngoại lệ là Educational Segment Tree Contest trên VNOJ. Hồi đó, em mới bắt đầu học Segment Tree. Các bài đã bao quát hết kha khá các kĩ thuật thường dùng, ý tưởng trong bài hay, đáng học hỏi.

Có một số kĩ thuật em đã biết từ trước khi làm Educational Contests. Em chủ động học các kĩ thuật ấy hay tình cờ bắt gặp trong các bài Codeforces và những OJ khác?

Thực ra đó là kết hợp của cả hai. Năm lớp sáu, đa phần em học thuật toán trong sách thầy Lê Minh Hoàng, kết hợp với việc tìm hiểu từ các anh chị chuyên đi trước. Thuật toán nào em thấy hay hay thì sẽ về Google và đọc bằng hết. Tuy nhiên, khi vốn thuật toán của em trở nên đầy đủ hơn, ít nhất ở trong khuôn khổ tỉnh em, thì không nhiều người biết một thuật toán nào mà em chưa nghe tới nữa. Em bắt đầu luyện bài và tìm ra những kĩ thuật hay ho từ ấy.

Mất bao lâu để em đọc một bài và đọc lời giải nếu như em bị mắc kẹt? Em nghĩ rằng phụ thuộc vào loại bài và độ khó của nó. Nếu em nghĩ từ 30 phút tới 1 tiếng, mà chả có hướng nào thì em sẽ đọc giải luôn. Ngược lại, em dành nhiều thời gian hơn cho một số bài nặng về quan sát (observation) như Atcoder, trung bình mỗi bài em dành 1 ngày cho nó.

Theo em, em là một người thích giải những dạng bài nào?

Điều này cũng khó nói, vì em không mạnh một phần nào đủ nhiều để có thể khẳng định được như thế. Tuy nhiên, đối với các bài không-phải-constructive và tham lam, khả năng giải được của em khá cao!

Em cảm nhận như thế nào phần thể hiện trong VNOI Cup Final của em?

Nếu phải dùng một từ để mô tả VNOI Cup Final, thì em sẽ dùng từ 'Choke'. Phong độ của em khi bước vào phòng thi không cao. Khi đọc đề bài 1, em nghĩ ra được solution ngay lập tức. Trong suốt ba tiếng tiếp theo, em dần dần tụt xuống tới tận rank 23 trên bảng xếp hạng. May mắn, có anh Võ Khắc Triệu cố gắng giải bài khó và chưa có điểm bài dễ. Em lúc đấy khá căng thẳng, overthinking (suy nghĩ nhiều) bài 2 và code bug liên tục trong suốt 3 tiếng đồng hồ. Em quyết định đi vệ sinh, và trong lúc đấy, em nghĩ rằng: “Ô, bài B dễ mà nhĩ!”. Thế là em quay về và AC câu B. Tâm lý thoáng hơn, em code một lèo AC câu C và cắn được 1250 điểm câu D. Một chút hình căn bản là đủ để AC câu E. Khoảng thời gian cuối cùng, em quay lại code bài D nhưng không thể code kịp.

Lý do nào khiến em quyết định tham gia VNOI Cup? Có phải do contest đó là kì thi duy nhất hiện lên trang VNOJ?



Lí do em thi VNOI Cup khá đơn giản. Sau VNOI Cup 2022, em lên Facebook chị Diễm Quỳnh (Admin VNOI), và thấy chị chụp áo VNOI Cup – phần thưởng cho các thí sinh xuất sắc nhất. Câu hỏi đầu tiên hiện lên trong đầu em: “Ừ, cái áo này đẹp quá! Làm sao để có áo này đây?”. Em quyết định thi VNOI Cup 2023, dành top 75 để có áo. VNOI Cup 2024 thì có khác biệt một chút. Cuối năm 2023, em trải qua một giai đoạn trầm cảm vì thành tích trong kì thi Tin Học Trẻ bảng B năm 2023. Em đã cài bug hơn một tiếng bài không quá khó trong tầm nghĩ của em, tuột mất giải Nhất. Em rất buồn trong thời gian dài, không có hứng thú tham gia bất kỳ kì thi nào. Em có một đứa bạn thân. Em với bạn ấy muốn chụp ảnh đôi. Em đã hứa với bạn rằng sẽ cố gắng lấy thêm một áo VNOI Cup nữa để có thể chụp ảnh đôi, vì sức bạn ấy không quá mạnh, hi vọng giành áo không thể lớn như em. Đây là lí do duy nhất em tham dự VNOI Cup 2024. Và với thật nhiều may mắn, em đã vào được chung kết.



Theo em, trong một kì thi, yếu tố tâm lý chiếm tỉ lệ như thế nào đối với kết quả? Diễn biến tâm lý của em diễn ra như thế nào qua các kì thi?

Em không phải là người có tâm lý tốt. Em thường đặt kì vọng rất cao và tự tạo áp lực để đạt được kì vọng đấy. Đặc biệt ở các kì thi online, em check bảng xếp hạng rất thường xuyên. Em không thích việc những bạn bình thường xếp dưới em đang đứng trên mình. Mặc dù tham gia nhiều kì thi, tâm lý của em chưa được cải thiện nhiều. Những phút đầu tiên, em sẽ bị căng thẳng vì lúc đó các bạn sẽ code bài dễ với nhịp độ rất cao. 20 phút đầu sẽ quyết định kết quả trong khoảng thời gian còn lại. Nếu em khởi đầu tốt, em sẽ hoàn thành tốt trong khoảng thời gian còn lại. Vì thế, giữ được bình tĩnh là điều rất cần thiết.

Được biết, em tham gia Codeforces từ giữa năm 2022. Cảm nhận của em trong contest đầu tiên như thế nào?

Trước contest đầu tiên trên Codeforces, em vừa tham gia kì thi Tin học trẻ toàn quốc. Tuy vậy, em vẫn cảm thấy rất hồi hộp khi lần đầu tiên làm bài trong contest lớn và độ khó cao như vậy. Hồi đó, em chưa biết khái niệm về div (div 2, div 3, etc.). Em đã thể hiện hết mình và làm không quá tệ! Có 6 bài Div 2 thì em làm đến tận bài C.

Trong những đề thi, bài tập mà em từng làm, em đã gặp bài nào mà em ấn tượng sâu sắc chưa? Vì sao?

Em đã từng gặp rất nhiều bài với cách giải vô cùng bất ngờ. Lập trình thi đấu là một thế giới vô cùng rộng lớn và chắc hẳn rằng sẽ có những thứ cực kỳ thú vị. Nhắc đến bài làm em cảm xúc nhất, em phải nói đến bài D Round 1 VNOI Cup 2024 (Nam châm). Em vẫn còn nhớ rất rõ đề câu đó. Đa số các bạn em đều theo hướng đi tham lam. Nó rất dễ để liên tưởng đến, nhưng cài đặt lại vô cùng khó. Với giới hạn N và toạ độ lớn, đáp án tối ưu chỉ có thể ở dưới dạng tiệm cận sau vô số thao tác. Lúc đấy, tâm trạng của em không được ổn định vì bài C em bị bug khá nhiều. Khi đọc bài D, em cũng nghĩ về tham lam. Như em đã nói, tham lam là điểm yếu của em. Do vậy, em nghĩ rằng: 'phải có gì đó sâu xa hơn tham lam!'.



Nam châm

Cho n nam châm trên trục tọa độ Ox , nam châm thứ i có tọa độ x_i . Các nam châm được đánh số từ 1 đến n theo tọa độ tăng dần, đồng thời có kích thước không đáng kể. Bạn có thể thực hiện phép toán sau với số lần tùy ý:

- Chọn hai nam châm i và j ($i < j$) và số thực t .
- Kích hoạt hai nam châm i và j với tham số t . Khi đó, nam châm i sẽ di chuyển đến tọa độ $x_i + t$, nam châm j sẽ di chuyển đến tọa độ $x_j - t$.
- Các nam châm không được phép vượt mặt nhau trong quá trình di chuyển. Nói cách khác, $t \leq \min(\frac{x_j - x_i}{2}, x_{i+1} - x_i, x_j - x_{j-1})$.
- Bạn sẽ nhận được điểm số t .

Hãy tìm tổng điểm số tối đa có thể nhận được sau khi thực hiện phép toán trên.

Input

Mỗi test gồm nhiều test case. Dòng đầu tiên của bộ dữ liệu chứa số nguyên dương t ($1 \leq t \leq 100$) là số lượng test case. Mỗi tá của mỗi test case như sau.

Dòng đầu tiên chứa số nguyên n ($2 \leq n \leq 2 \cdot 10^5$) — số lượng nam châm.

Dòng thứ hai chứa n số nguyên $x_1, x_2, \dots, x_n$ ($0 \leq x_1 < x_2 < \dots < x_n \leq 10^9$) — tọa độ của các viên nam châm.

Dữ liệu đảm bảo tổng n trong tất cả các test case không quá $2 \cdot 10^5$.

Output

Với mỗi test case, in ra một số thực duy nhất là tổng điểm lớn nhất có thể nhận được.

Định nghĩa J là đáp án của bạn tổ chức, P là đáp án mà ban đưa ra. Đáp án của bạn sẽ được cho là đúng nếu như sai số tương đối hoặc tuyệt đối so với đáp án của ban tổ chức không vượt quá 10^{-6} , hay $\frac{|P - J|}{\max\{1, |J|\}} \leq 10^{-6}$.

Bài D - Vòng loại thứ nhất VNOI CUP 2024

Em tự hỏi rằng bài toán này có thể chuyển về dạng đơn biến hay không? Vì yếu tham lam, em bắt tay vào làm theo hướng đó luôn. Và may mắn thay, nó chính là solution chuẩn của BTC! Khi xét tổng của các cặp nam châm, số ấy là một hàm đơn biến, có thể tính được đáp án cực kỳ nhanh. Em rất bất ngờ, chả lẽ đề câu D VNOI Cup lại dễ thế này à!? Không dễ nhìn, nhưng ý tưởng của một bài D VNOI Cup chỉ code 20 dòng được! Em một đấm AC bài đó, và là người thứ hai giải xong bài này (sau `neko_nyaaaaaaaaaaaaaaaaaaaaaaaaaaaa`).

Em đã làm rất nhiều bộ đề, chắc hẳn em cũng biết đề thi HSGQG³⁸ các năm gần đây rất khó. Em đã thử làm một bộ đề nào đó chưa? Em đánh giá đề HSGQG như thế nào so với các kỳ thi em từng tham gia?

Em chưa từng bấm giờ làm thử đề HSGQG bao giờ. Tuy nhiên, đây là một bộ đề khá khó. Em từng đọc đề VOI 2022 với bài 1 là một bài Quy hoạch động rất ảo. Em có nói chuyện với một anh năm đấy được Nhì VOI, trình bày về ý tưởng “tham lam mà quy hoạch động” của mình. Nghe xong, anh bảo: ‘Ừ đúng rồi em. Bài này anh làm bốn cái bao lồi’. Đề VOI rất khó, và từ đấy đến nay em chưa bao giờ làm lại.

Cách luyện tập của em yêu cầu phải AC luôn, không có subtask hay điểm lẻ. Em đã bao giờ làm các đề OI tương tự như VOI của các nước khác chưa (COCI, USACO, ...)?

Em có làm một ít đề. Có một thời, em luyện rất nhiều đề CEOI³⁹ và EGOI⁴⁰. Lúc em luyện những bài đấy, em dựa vào contest trên `vjudge`⁴¹ và không có cố gắng cần điểm. Do vậy, kĩ năng đoán test của em không quá tốt. Lần gần nhất em làm thử một bộ đề OI là vài tháng trước với đề IOI⁴² 2020. Bài 2 là một bài constructive không quá khó, em AC trong chưa đến 1 tiếng làm. Qua bài 3, sau khi suy nghĩ thì em nhìn ra được thuật greedy (tham lam) và cũng AC được. Nhưng bài 1 lại là bài rất khó, và em cần test rất tệ. Trong suốt 3 tiếng đồng hồ, em chỉ cài được 5/100 điểm. Em không xem bảng xếp hạng của kì thi. Hai bài 2, 3 không khó và các anh chị huy chương Bạc trở lên đều AC bài này. Bài 1 là bài có mức phân hoá cao, và em không làm được tốt nên chỉ được mức huy chương Bạc. Nếu em cần test tốt hơn thì em đã có vị trí cao hơn đôi chút.

Để cải thiện kĩ năng cần test, em nghĩ mình phải luyện tập những gì?

Kĩ năng cần test không phải là thế mạnh của em. Có thể em sẽ thử làm nhiều đề cần test và bấm giờ. Em nhớ một lời khuyên của anh Phạm Văn Hạnh, rằng nên làm lại các đề và bấm giờ làm bài. Từ đó, mình có thể rút ra được lỗi sai trong chiến thuật của mình.

³⁸HSGQG: là viết tắt của kỳ thi Học sinh giỏi Quốc gia.

³⁹<http://ceoi.inf.elte.hu/>

⁴⁰<https://ego.org/>

⁴¹<https://vjudge.net/>

⁴²IOI: Kỳ thi Olympic Tin học Quốc tế.



Em là một trong những người trẻ tuổi có khả năng đạt rating cao trên Codeforces. Em có bao giờ đi đọc code của những người khác để học hỏi không?

Nếu có thời gian, việc đọc code sẽ rất hữu ích. Tuy vậy, em tự ra cho mình rất nhiều bài, nên thời gian của em khá hạn hẹp để đọc code của người khác. Nhưng nếu code mình không tối ưu, đọc code tham khảo sẽ là một ý hay. Ngoài ra, code của các anh chị đã bỏ CF thì thường không đẹp cho lắm, nên em không có hứng thú lắm.

Các bạn 'đỏ đen' CF thường sẽ có phong cách riêng. Như Jiangly thì code rất đơn giản, chân phương, không cần sử dụng template. Tourist chuẩn bị cho mình template rất đầy đủ và tận dụng tối đa sức mạnh của template để đạt được lợi thế. Khi em đọc code của Benq, em sẽ cảm thấy cực kì khó hiểu. Em tự định nghĩa phong cách code của em như thế nào?

Cách code của em là build-from-scratch, em thường code lại tất cả mọi thứ từ đầu. Em không dùng template, đặc biệt là vòng loại VNOI Cup, em chả code sẵn cái nào luôn. Những kì thi mà em tham gia, yêu cầu thuật toán sẽ ít hơn rất nhiều so với yêu cầu về tư duy. Những bài cần dùng thuật toán khó đồng nghĩa với việc tư duy của nó phải khó hơn gấp nhiều lần, vượt quá tầm với của em. Các bài em làm được, đa số đều không cần sự cầu kì về mặt kĩ thuật code. Do vậy, em không nghĩ tới việc làm code sẵn. Tuy nhiên, với những kỳ thi như VNOI Cup Final hay ICPC⁴³ Regional Hà Nội, em nghĩ rằng việc chuẩn bị template là 'có còn hơn không'. Chuẩn bị xong thì em chả dùng lần nào, từ VNOI Cup Final, ICPC Quốc gia đến ICPC Regional.



AI đang ngày càng mạnh hơn, tạo nên sự so sánh rằng AI đã đạt trình độ Đỏ CF (2300 – 2400). Theo em, với việc AI đã có thể giải được các bài Lập trình thi đấu, em sẽ xem AI như là một công cụ hay hoàn toàn không sử dụng nó?

Em là một người theo tư tưởng truyền thống. Em không sử dụng AI trong CP và các công việc hằng ngày. Lập trình thi đấu không phải là đọc code sẽ tự động lên trình, mà phải trải qua khoảng thời gian suy nghĩ, lập luận, mới có thể tiến xa hơn. Em tin rằng trong Lập trình ứng dụng, AI sẽ hỗ trợ rất nhiều. Ví dụ về thiết kế web, ai cũng biết rằng code front-end cực kì mệt mỏi – mỗi nút cần một code riêng, phải gửi thông tin về back-end như thế nào, thiết kế UI ra sao, ... Nắm vững kiến thức cho front-end không hề khó, nhưng code dài và tốn thời gian. Em tin AI là công cụ tuyệt vời vì nó có khả năng làm những công việc ấy nhanh hơn rất nhiều.

Em đang trong giai đoạn ôn thi vào chuyên. Vậy em có thể chia sẻ nguyện vọng trường trong tương lai của em được không?

Em nghĩ rằng nguyện vọng cao nhất của em là Trường THPT Chuyên Khoa học Tự nhiên Hà Nội. Sau một khoảng thời gian ngắn theo học trên đây, em tin rằng đó là một môi trường vô cùng tốt để phát triển thêm trên con đường Lập trình thi đấu của bản thân. Hiện tại, em có giải Nhất Tin Học Trẻ Toàn Quốc bảng B. Nếu quy định không thay đổi, em hoàn toàn có thể được xét tuyển thẳng vào trường THPT Chuyên Khoa học Tự nhiên. Tuy vậy, đây là môi trường rất cạnh tranh, và thi chuyên Tin KHTN sẽ phải làm toán chuyên, vốn không phải là thế mạnh của em. Do vậy, nguyện vọng 2 của em sẽ là trường THPT chuyên Lương Văn Tụy quê em.

⁴³ICPC: International Collegiate Programming Contest - Cuộc thi Lập trình Quốc tế lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu.



Em dành 4 tiếng mỗi ngày cho môn Tin học. Những khoảng thời gian còn lại, em phân bổ cho các môn ngữ văn, tiếng Anh, ... như thế nào?

Em tự nhận rằng mình là một người khá 'vô kỷ luật', em thật sự không có thời khóa biểu nào trong ngày. Em hứng cái gì thì em sẽ học cái đấy. Làm tin chán quá thì lời toán ra đọc, đọc chán thì lại chơi, chơi chán thì lại ngồi code! Trên lớp, em cố gắng tranh thủ chép hết bài trong các môn phụ, dành thời gian suy nghĩ các bài khó. Buổi chiều em được nghỉ, nên em sẽ học theo ý thích từ 2 tới 5 giờ chiều. Buổi tối, em học từ 8 tới 11 giờ tối rồi đi ngủ.

Em có cơ hội tiếp xúc Lập trình thi đấu từ rất sớm. Nếu không chọn môn Tin, em nghĩ hướng đi của em sẽ như thế nào?



Khả năng cao nhất của em chắc vẫn ở môn Toán. Nhưng em không quá mạnh Toán, nên có lẽ em sẽ đổi hướng qua các môn tự nhiên khác như lý, hóa, ... Với các môn khác, em sẽ không cạnh tranh bằng môn Tin. Em cảm thấy may mắn vì đã chọn Tin làm môn chuyên của mình.

Ngoài việc học, em còn sở thích nào không?

Bản thân em không chơi game, nên ngoài giờ, em sẽ chơi cầu lông hoặc cờ vua. Cầu lông em chơi khá ổn và khá nhiều. Có một thời gian em đi học ở Hà Nội, mỗi khi học xong lúc 4h chiều, em sẽ chơi tới tận 7h tối rồi mới về ăn cơm. Em chưa đi thi cầu lông bao giờ. Trường chọn đội tuyển cầu lông từ các đại diện lớp. Lớp em có hai bạn, một bạn dân bán chuyên, một bạn từng HCB Tỉnh, cả hai em đều không đánh lại. Do đó em không có cơ hội cạnh tranh nhiều. Em biết đánh cờ vua, từng tham gia Hội khỏe Phù Đổng cờ vua nhưng không có giải vì thua hệ số phụ.

Sau khi đậu chuyên, em có định hướng tương lai nào không?

Em sẽ cố gắng đậu đội tuyển và thi HSGQG. Nếu may mắn, em có thể thi TST⁴⁴, APIO⁴⁵ và IOI! Em không có quá nhiều dự định trong tương lai, một phần vì cá nhân em xuất thân từ vùng mà phong trào của ngành Tin học nói chung vẫn chưa phát triển, em cũng chưa được biết nhiều về các hướng đi tiềm năng. Do vậy, em đang tập trung cho các dự định ngắn hạn đã.

Em đã bao giờ cảm thấy mất động lực, "chán học" môn Tin chưa?

Cá nhân em chán học Tin rất nhiều. Học từ 4-6 tiếng một ngày, không tránh khỏi những suy nghĩ tiêu cực khi trình độ không nhích lên. Em lấy động lực từ gia đình em với sự hỗ trợ rất lớn trong việc học tập của em. Em mong rằng mình có thể trở thành lập trình viên thật tài năng để chăm sóc được gia đình em sau này.

⁴⁴TST: Kỳ thi tuyển chọn đội tuyển dự thi Olympic Tin học Quốc tế, hay còn được biết đến là Vòng 2 thi chọn Học sinh giỏi Quốc gia.

⁴⁵APIO: Kỳ thi Olympic Tin học Châu Á -- Thái Bình Dương.



Anh cảm thấy rằng em có rất nhiều trải nghiệm đáng quý để chia sẻ cho quý độc giả. Em có lời khuyên, lời chúc nào cho các bạn độc giả nhân dịp Tết này không?

Em nghĩ rằng lời khuyên giá trị nhất của em chính là: “Hãy nhớ tại sao mình bắt đầu”. Chỉ khi nhớ được những lí do bắt đầu, mình mới tạo ra động lực để vượt ngàn chông gai, chiến thắng bản thân mình.

Rất cảm ơn em đã nhận lời mời tham gia hôm nay và chia sẻ nhiều câu chuyện, kinh nghiệm hữu ích. Năm mới sắp đến, ban biên tập chúc em thành công trong kì thi tuyển sinh sắp tới, học tập tốt và đạt được nhiều thành công hơn nữa trong hành trình Lập trình thi đấu!





Phỏng vấn Nguyễn Tấn Sỹ Nguyên

Interviewer: Nguyễn Trung Quân – Ban biên tập Tạp chí VNOI



Em chào anh, được biết anh là một trong những người tham gia lập trình thi đấu từ rất lâu và vẫn tiếp tục tham gia cho đến thời điểm hiện tại, VNOI mong muốn qua buổi phỏng vấn này có thể đem đến cho độc giả những cái nhìn sâu sắc hơn về trải nghiệm của anh đối với bộ môn này.

Trước tiên, anh có thể giới thiệu về bản thân mình được không?

Anh tên là Nguyễn Tấn Sỹ Nguyên, anh là một trong những người cũng thuộc dạng là "thi thố" từ lâu rồi, anh năm nay 32 tuổi. Anh đi thi HSGQG⁴⁶ cũng từ thời 2009 – 2010, thì sau đó năm 2011 anh được thi quốc tế, rồi sang thời đại học thì anh cũng được đi thi ICPC⁴⁷ World Finals⁴⁸ hai lần, năm 2013 và 2016. Sau đó là anh tốt nghiệp và đi làm, gần đây anh cũng có train và thi lại các kỳ thi Lập trình thi đấu thì cũng được vào Vòng chung kết của VNOI Cup ba lần.

Trong quãng thời gian học cấp 3 thì anh có theo học tại trường Phổ Thông Năng Khiếu, anh có thể kể về quãng thời gian khi anh theo học tại ngôi trường này được không? Và nếu so với điều kiện học tập bây giờ thì theo anh thấy có sự khác biệt như thế nào?

Ừm! Ngày xưa lúc anh bắt đầu học thì anh thi vào môn Toán chứ anh cũng không có kiến thức gì về Tin. Sau đó lúc anh vào lớp 10 thì tụi anh cũng phải học lại từ đầu những kiến thức cơ bản về Tin. Nhờ có một đợt là anh cũng được may mắn được chọn vào đội tuyển để đi thi Olympic 30/4 của trường á thì thì sau này đó thì anh mới bắt đầu là anh kiểu tập luyện nhiều hơn. Rồi cũng tìm được cái trang SPOJ hồi đó thì cũng lên đó làm bài của mọi người thì mới bắt đầu là tiến bộ nhiều.

Thời của anh thì tụi anh cũng là khi mà lên đầu năm lớp 11 mà khi chọn đội tuyển vào đội tuyển thi quốc gia xong rồi thì sau đó thì các bạn đội tuyển thì cũng được dành tất cả thời gian để luyện tập và cũng không có phải học ở các môn khác ở trường thì anh nghĩ là trường cũng đã cố gắng tạo điều kiện khá là tốt cho tụi anh để tụi anh có thể là tập trung hết sức và luyện tập để dành tập trung trong kỳ thi Quốc gia. Anh nghĩ là cái điều kiện đó so với bây giờ thì cũng khá là tương đồng, tại vì anh cũng có nói chuyện với các bạn khóa sau này thì điều kiện đó vẫn còn được giữ. Nhưng có một cái đặc biệt anh thấy là sau này mọi người có thêm những cái cơ hội để tham gia những đợt tập huấn ở những nơi khác, như là các tỉnh khác như miền Bắc, miền Trung hoặc là tham gia những cái đợt thi thử thì anh nghĩ là những dịp như thế như thế sẽ giúp cho các bạn có thêm kinh nghiệm khi đi thi. Tại vì dù tụi em có luyện tập nhiều nhưng mà cái lúc em vào phòng thi thì cái cảm giác cũng sẽ khác và sẽ liên quan đến nhiều thứ như chiến thuật và tâm lý của mình. Anh nghĩ đó là một trong những điểm mà sau này có thể các bạn sau này có được những thuận lợi hơn so với thời của tụi anh. Thời anh thì thường là những cái thi thử đó nó diễn ra online nhiều hơn, thường cũng là tự phát từ cộng đồng VNOI thời đó thôi. Kiểu các anh lớn mà có thời gian các anh sẽ cố



⁴⁶HSGQG: là viết tắt của kỳ thi Học sinh giỏi Quốc gia.

⁴⁷ICPC: International Collegiate Programming Contest - Cuộc thi Lập trình Quốc tế lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu.

⁴⁸World Finals: Cuộc thi ICPC cuối cùng trong một mùa giải bao gồm những đội xuất sắc nhất của các trường đại học trên thế giới vượt qua vòng loại vùng (Regional contest)



gắng chuẩn bị đề cho các bạn thi thử, thì đó cũng là một dịp tốt để cọ xát nhưng mà dĩ nhiên nếu mà so với việc được đi thi trực tiếp lên đến nơi và phòng máy thì nó sẽ không có bằng được.

Theo anh thì đâu là bí quyết góp phần giúp anh có được HCB IOI⁴⁹ năm 2011?

Anh nghĩ SPOJ là cái xuất phát điểm của anh. Trong thời gian mà anh học thì anh cũng làm rất nhiều, nhưng anh nghĩ như vậy chưa đủ mà anh cũng phải luyện thêm những cái contest trên mạng, thì hai contest mà anh thấy khá là gần với lại cái kỹ năng để mình đi thi quốc gia và thi quốc tế thì nó sẽ là COCI⁵⁰ của Croatia với lại USACO⁵¹ Mỹ, đó là hai series kỳ thi mà chất lượng khá là tốt. Ngoài ra thì thời của tụi anh có TopCoder, và lúc đó thì Codeforces⁵² cũng vừa bắt đầu xuất hiện.



Từ trái sang: Nguyễn Hoàng Yến, Lê Khắc Minh Tuệ, Nguyễn Vương Linh và Nguyễn Tấn Sỹ Nguyên tại kỳ thi IOI năm 2011

Anh đánh giá là mặc dù những kỳ thi trên các nền tảng đó cũng tốt nhưng mà lại không sát với thể thức thi quốc gia cho lắm. Vì vậy, nếu các bạn làm thì cũng cũng tốt nhưng thời gian mà có hạn thì mình nên tập trung vào những kỳ thi có hình thức OI. Khi mình thi những kỳ thi đó, mình sẽ học được cách phân bổ thời gian và giúp tạo ra phản xạ làm những subtask của bài, vì thực tế sẽ có những bài mình không thể làm trọn điểm được và khi đó mình sẽ phải chuyển qua chiến thuật cố gắng ăn điểm từng phần của bài. Đây cũng là ưu điểm của kỳ thi hình thức OI khi giúp cho mình rèn luyện và thử nghiệm các chiến thuật. Ngược lại, nếu luyện bài trên các Online Judge như SPOJ thì mình không có cái giới hạn thời gian như vậy, kiểu như mình có thể cầm một bài và nghĩ từ ngày này sang

ngày khác, theo anh thì cách này cũng tốt, nhưng mà nó sẽ là luyện một cái kỹ năng khác chứ không phải những kỹ năng mà anh đã đề cập.

Anh cũng là một người tham gia Codeforces từ những ngày đầu tiên. Theo anh thấy contest Codeforces ngày xưa và contest Codeforces bây giờ có sự khác nhau trong cách ra đề không?

Về xu hướng ra đề của Codeforces thì tháng 12 năm trước, mọi người cũng hay phàn nàn là meta toàn constructive. Sau đợt đó thì đề Codeforces bây giờ cũng đỡ một chút và có vẻ là cũng cân bằng hơn. Nhưng mà anh nghĩ cái đội ngũ Coordinator của Codeforces chất lượng cũng hơi không có ổn định lắm, cũng người này người kia nên dẫn đến là những cái vòng thi có chất lượng bài không có đồng đều, có những vòng bài hay có những vòng có bài dở bị chê lên chê xuống. Anh nghĩ là cái đó thì mình cũng chấp nhận thôi tại vì mọi người làm miễn phí cho mình thi nên như vậy thì cũng là vui rồi. Ngày xưa anh nghĩ dạng bài nó sẽ truyền thống hơn, nó không có dùng những cái kiểu thiên về tư duy hay constructive nhiều như bây giờ, vì có thể ngày xưa thì nó vẫn còn khá là nhiều cái dạng mà mình chưa có tận dụng hết, bây giờ thì anh nghĩ là tới thời điểm mọi người cái gì cũng đã biết rồi, mình giải được rồi thì bây giờ những cái bài mới nó phải có một cái gì đó sáng tạo thì mới được cho vào đề, dẫn đến việc là bây giờ là cái kiểu bài của Codeforces nó sẽ có những bài yêu cầu về tư duy như vậy.

Theo em được biết thì trên Codeforces thì có những người trong top 10 Codeforces cũng là thần tượng của rất nhiều người, không biết anh có thần tượng người nào trên Codeforces hay trên các OJ khác không ạ?

Thật ra thì anh không thần tượng ai cả. Nói chung là dù anh cũng thích xem livestream mọi người thi Finals, nhưng anh không đặc biệt cổ vũ ai cả. Dạo này anh đọc code thì anh thấy thích code của Jiangly, bạn viết code khá là dễ hiểu. Còn đọc code của Tourist thì cũng được, không tệ lắm. Nhưng mà nhiều khi chỉ đọc vậy thôi chứ không biết bên kia đang code gì luôn.

⁴⁹IOI: Kỳ thi Olympic Tin học Quốc tế.

⁵⁰<https://hsin.hr/coci/>

⁵¹<https://usaco.org/>

⁵²Codeforces: một trang web tổ chức các cuộc thi lập trình thi đấu với các dạng đề, bài tập đa dạng.



Với kinh nghiệm dày dặn trong việc luyện tập và thi các kỳ thi khác nhau, theo anh thấy thì kỳ thi HSGQG so với IOI thì có nhiều sự khác biệt nào và cách tiếp cận bài toán trong hai kỳ thi này có khác nhau không?

Theo anh thấy thì kỳ thi HSGQG của mình khác khá là nhiều. Nói chung là anh cảm thấy nó khác với tất cả những kỳ thi IOI của các nước khác luôn chứ không chỉ là IOI. Tại vì cái quan trọng nhất là mình phải thi offline. Anh đánh giá là cái thể thức thi offline là một cái thức rất là khắc nghiệt, bởi vì có những cái lỗi căn bản mà nếu em gặp phải là em sẽ bị 0 điểm luôn một bài. Vì vậy các kỹ năng để đi thi đi VOI thì nhiều khi nó khác khá là xa so với kỹ năng để thi IOI. Thời của anh thì thi vòng hai TST⁵³ cũng vậy, nhưng mà theo anh biết sau này thì thi TST đã đổi lại thể thức cũng khá giống IOI rồi, mọi người được biết điểm của mình khi nộp bài luôn thì anh nghĩ nó đã bước tiến khá là lớn rồi. Hy vọng sau này trong tương lai gần mình cũng có thể làm được điều đó với lại cái kỳ thi HSGQG, nếu điều đó xảy ra sẽ làm cho kỳ thi HSGQG nước ta có thể bám sát với thể thức thi IOI của thế giới hơn và giúp cho các bạn luyện được kỹ năng đó từ giai đoạn đầu chứ không phải đợi tới vòng 2 mới có cơ hội để luyện.

Như anh cũng biết là kỳ thi HSGQG vừa diễn ra. Thì đề HSGQG cũng đã được đăng công khai trên các nền tảng rồi, anh cũng có đọc qua thì anh đánh giá đề VOI năm nay như thế nào, so với đề thi ngày xưa mà anh đã từng thi?

Nếu so với ngày xưa thì anh nghĩ ngày xưa đề nó dễ hơn nhiều. Nói chung ngày xưa lúc nào cũng sẽ có một bài dễ mà đọc vào nghĩ tầm 5-10 phút là sẽ ra. Còn sau này, ví dụ như đề năm ngoái thì khá là khó. Còn đề năm nay có vẻ là mấy thầy đã nương tay hơn, cũng bắt đầu có những bài các bạn có thể làm được, nhưng mà gọi là có thể làm được chứ cũng phải code tương đối.

Đề sau này anh có cảm giác là mình bắt đầu bám sát cái syllabus của IOI nhiều hơn vì bắt đầu có rất là nhiều subtask cho mọi người làm. Nhưng mà với cách anh suy nghĩ thì nếu mà đã cho nhiều subtask như vậy thì thời gian thi cũng nên tầm 4-5 tiếng, và cũng nên có chấm bài online thì nó sẽ sát hơn, chứ cho nhiều subtask xong lại bắt mọi người nộp offline thì nó lại có nhiều vấn đề như là mình không biết điểm, mình cũng không biết là cái subtask đó mình code có đúng hay không, thì nhiều khi là không biết là có nên code subtask này tiếp hay không, các bạn đôi lúc phải đấu tranh tâm lý rất nhiều vì hơi hơi mâu thuẫn với nhau một chút. Theo hướng tích cực anh nghĩ là chất lượng đề đã khó hơn so với thời anh thì cũng là một cái tốt rồi. Anh thấy năm nay mọi người có nhận xét là có một cái bài giống như một bài đã từng xuất hiện trên Codeforces, anh nghĩ là thật ra cái chuyện đó cũng không sao cả tại vì khi vào phòng thi thì mọi người cũng đâu có truy cập được mạng, cũng phải tự suy nghĩ thôi. Còn những ai mà may mắn mà làm bài đó rồi thì coi như bạn ấy may mắn, bạn ấy luyện tập nhiều thì bạn ấy xứng đáng để đến để làm được bài đó. Chứ còn bảo là để yêu cầu là tất cả các bài phải là bài mới hết thì anh nghĩ là nó cũng tương đối khó, nghĩ ra một bài chất lượng để cho thi HSGQG có nhiều subtask thì nó không phải là một điều đơn giản, cho nên anh thấy cái chuyện nó trùng bài nó bình thường thôi, miễn sao là tất cả mọi người thi với nhau công bằng thì cuối cùng ra kết quả ai được điểm cao thì nó vẫn là xứng đáng.



⁵³TST: Kỳ thi tuyển chọn đội tuyển dự thi Olympic Tin học Quốc tế, hay còn được biết đến là Vòng 2 thi chọn Học sinh giỏi Quốc gia.



Theo em được biết thì anh từng có thời gian hoạt động cho VNOI ngày xưa, theo anh thấy các kỳ thi được tổ chức bởi VNOI bây giờ như nào?

Ngày xưa tụi anh cũng chưa có nhiều kinh nghiệm để làm mấy cái đấy lắm, tụi anh hồi đấy muốn đóng góp cho cộng đồng, và suy nghĩ đến việc ra đề rồi tổ chức những vòng online. Nhưng nếu so sánh với cái quy mô của VNOI Cup bây giờ thì anh nghĩ bây giờ mọi người đã tiến được khá là xa so với ngày xưa, và cái anh đánh giá cao nhất là việc mình có thể tổ chức được một cái vòng chung kết onsite và quy tụ được không chỉ là các bạn thí sinh đi thi mà còn mở cho các bạn thí sinh ở bảng mở rộng nữa; những cái như là livestream rồi quay màn hình thí sinh các thứ rất chuyên nghiệp, anh nghĩ đây là một cái không thể tưởng tượng được với một cộng đồng như là cộng đồng của mình có thể tự làm được điều đó. Anh nhớ ngày xưa cũng có một đợt VNOI tổ chức VNOI Marathon có onsite chung kết. Nhưng mà cái onsite đó là nhờ bên phía công ty Cốc Cốc, họ cho mượn văn phòng của họ rồi cho các bạn đến tham gia. Và hồi đó thì anh cũng không nhớ là có tài trợ vé máy bay hay là chỉ dành cho các bạn ở miền Bắc nhưng mà anh nghĩ với quy mô bây giờ nhìn lại so với ngày xưa thì mình đã làm được tốt hơn rất là nhiều. Và anh rất hy vọng là phong trào này sẽ tiếp tục được duy trì, không chỉ là VNOI Cup mà kể cả là những cái kỳ thi tại Việt Nam diễn ra hàng năm, để cho các bạn cấp Ba rồi các bạn sinh viên có thêm những cơ hội luyện tập, để giúp cho cộng đồng tin học của Việt Nam phát triển nhiều hơn.

Như em được biết thì VNOI Cup hướng đến việc thí sinh có những trải nghiệm tốt nhất khi đi thi, theo anh thấy điều này có phải là điểm mới so với nhiều kỳ thi khác hay không?

Đúng, cái này anh đồng ý. Anh thấy qua từng mùa thì các bạn bên ban tổ chức cũng đã cầu thị hơn và cũng đã thể hiện những thay đổi đó. Như mùa đầu thì mọi người phải in template ra, nhưng qua tới mùa 2 được phép copy code có sẵn rồi. Sau đó còn được ban tổ chức hỗ trợ cài thêm các IDE, plugin tùy theo sở thích của mọi người nữa. Nói chung đúng như là ban tổ chức nói, chỉ thiếu cái là mở mạng cho mọi người thôi! Việc thi cảm giác bây giờ như mọi người thi ở nhà luôn rồi, anh nghĩ đây là điểm để giúp cho các thí sinh có được cái điều kiện là thoải mái nhất như là mình đang dùng máy cá nhân, nó là một cái rất là hay.

Năm 2024 chứng kiến sự đột phá của anh trong thứ hạng tại VNOI CUP, trong quãng thời gian này anh có luyện tập hay là có phương pháp nào không?

Thực ra thì anh thấy anh luyện giữa ba năm nó cũng ngang ngang nhau, với lại nếu mà so về rating ở trên Codeforces thì năm vừa rồi anh đi thi thì rating còn thấp hơn nữa. Anh nghĩ là một trong những cái yếu tố ảnh hưởng nhiều đến cái kết quả của anh là cái phần mà anh set up cái IDE của anh, thì anh set up IDE rất là cẩn thận và anh cố gắng mô phỏng lại hết tất cả những gì mà anh hay dùng ở nhà. Lúc đó anh quen nên anh làm mọi thứ nhanh hơn.



Anh Nguyễn Tấn Sỹ Nguyên (ở giữa) trong buổi giao lưu tại VCK VNOI CUP 2024

ảnh hưởng khá là nhiều.

Theo anh thì yếu tố tâm lý có ảnh hưởng nhiều đến kết quả kỳ thi hay không?

Anh nghĩ đó là một trong những cái quan trọng nhất, kể cả thi online hay offline. Bởi vì như anh thì anh cũng rất là hay mắc cái lỗi là nhiều khi là có một cái bài dễ xong rồi anh bị kẹt chỗ nào đó nghĩ không được nên anh cứ ngồi làm bài đó cho tới hết giờ. Anh cũng phải luyện tập rất nhiều để vượt qua

Với lại thêm một cái nữa là anh khắc phục được việc vướng các bài dễ lâu quá, lúc thi mình cũng có được một cái tâm lý ổn định. Như năm đầu tiên, có một bài dễ cản anh ngồi hết từ đầu đến cuối, cuối cùng anh cũng không giải được bài A nên nó cũng ảnh hưởng đến tinh thần khi thi. Anh nghĩ là về kỹ năng có thể là không có thay đổi nhiều nhưng sẽ có những yếu tố nhỏ nhỏ như về set up IDE thì sẽ ảnh hưởng đến mình. Như năm đầu tiên anh thấy anh toàn bấm nhầm phím không à, tại vì bình thường anh dùng Mac á cho nên là anh dùng phím CMD, trong khi đi thi thì phải dùng phím Ctrl xong anh cứ bị nhầm, cũng



cái suy nghĩ đó, kiểu là đồng ý bài này mình không được thì mình chấp nhận coi như mình mất điểm bài đó, mình làm bài tiếp theo, một lúc sau cuối giờ mình quay lại nghĩ tiếp, biết đâu mình lại nghĩ ra được một hướng nào khác. Cái yếu tố tâm lý nó quan trọng ở chỗ đó. Giống như làm VNOI Cup thì nhiều khi là làm những cái bài dễ xong rồi làm không được thì tâm lý mình cũng bắt đầu bị ảnh hưởng nhiều. Thật ra cái cái hay của việc em không biết bảng rank thì là nhiều khi em không làm được bài đó thì em cũng không biết nó dễ hay nó khó thì em cứ kệ nó mà làm bài khác, thì em cũng đỡ lo lắng hơn. Anh nghĩ đó là một trong những cái mà các bạn sẽ cần phải luyện tập khi mà đi thi, đặc biệt các kỳ thi lớn như IOI. Anh thấy điểm IOI dạo này khá là thấp chứng tỏ đề IOI dạo này khá là khó. Tại vì khi mình thi mình thấy là bài này mình được có 10 điểm, 15 điểm xong mình sợ. Nhưng mà nhiều khi bài đó khó, phần lớn mọi người ai cũng 15, 20 điểm thôi. Nên là mình phải luyện tập tâm lý là bao nhiêu điểm thì nó không quan trọng, bây giờ mình cứ tập trung vào có bài nào làm được thì mình làm, có subtask nào làm được mình làm, nếu mà mình cứ dựa vào cái tâm lý đó thì mình sẽ thể hiện được cái cái phong độ tốt nhất của mình. Còn nếu mà mình lo lắng ít điểm quá rồi xong mình cứ phải cố gắng mình làm thêm thì mình tự nhiên bỏ đi những bài mình có thể làm được. Ví dụ như anh nhớ đợt IOI năm rồi thì đoàn mình cũng có bạn nào đó bỏ bài dễ của ngày 2 nên cũng hơi tiếc.

Anh cũng là một người từng thi rất là nhiều khi rất là nhiều cái thi mà kiểu theo hình thức chấm subtask, như là VNOI Cup hay IOI. Anh có xu hướng nghĩ bài theo subtask hay nghĩ một phát AC luôn?

Thường là anh sẽ đi ngược, anh sẽ nghĩ cách giải tổng quát trước, nghĩ không ra tới lúc bắt đầu mà cảm thấy là anh không đủ thời gian bài đó nữa thì anh sẽ nghĩ theo hướng subtask, đó là lúc anh thi VNOI Cup. Còn IOI thì lúc đi thi anh làm theo kiểu ăn subtask, năm của anh có 2 bài dễ, còn những bài kia sẽ khá là khó thì nếu mà mình xác định cái khả năng của mình là như thế nào và mình ăn chắc những cái subtask trước thì mình cũng sẽ cảm thấy tự tin hơn. Chứ còn nếu mà mình ngồi để làm full cái bài đó xong rồi chẳng may là bị thiếu sót một vài chỗ rồi cuối cùng mình không có điểm nào thì nó là một cái rất là dở, cho nên anh nghĩ là mình sẽ phải đánh giá được khả năng của mình và đề bài, với đề bài đó mình sẽ bắt đầu mình chọn chiến thuật phù hợp. Thi IOI khó hơn ở chỗ là mình không biết điểm của mọi người, còn VNOI Cup thì mình biết bảng điểm của mọi người, ai làm bài nào thì nó vẫn dễ hơn.

Được biết là lúc lên đại học thì anh có dành một năm ở Trường đại học Khoa học tự nhiên và sau đó anh mới chuyển qua đại học Quốc gia Singapore (NUS) để theo học. Ở quãng thời gian anh học tại Trường đại học Khoa học tự nhiên thì anh có tham gia những kỳ thi lập trình thi đấu không hay là anh sẽ chỉ có dành thời gian để nghỉ ngơi và học trước kiến thức cho năm sau này?

Có, lúc năm anh ở Trường đại học Khoa học tự nhiên anh cũng có thi ICPC luôn, anh cũng được tham gia thi tại hai Regional nhưng mà kết quả của anh năm đó thì không có tốt lắm. Các mức rank team anh thường đạt được nằm ngoài khoảng top 10. Anh cũng có đi thi Olympic Tin học Sinh viên, năm đó có một điều anh cũng không hiểu tại sao là anh cũng làm bài không được tốt lắm nhưng cũng được CUP vàng.

Theo anh thấy là Trường đại học Khoa học tự nhiên cũng là một trong những trường đầu tư khá là tốt vào cái việc luyện tập và tổ chức cho các sinh viên trong trường đi thi ICPC. Đó cũng là một cái mà anh cũng có nghiên cứu khi mà anh chọn trường.

Sau này khi anh qua NUS thì anh có hai lần tham dự ICPC World Finals, nhưng đó không phải những năm đầu tiên phải không?

Đúng rồi, anh thi năm 1 và năm 4, năm đầu và năm cuối.

Theo anh thấy thì cách tổ chức training ICPC ở KHTN với NUS nói riêng và Việt Nam với Singapore nói chung có sự khác nhau như thế nào và bên nào



Từ trái sang: Nguyễn Tấn Sỹ Nguyên, Nguyễn Thành Trung, Nguyễn Hùng Tâm tại kỳ thi ICPC WF năm 2016



có sự đầu tư hơn? Cái này cũng hơi khó so sánh, theo

anh nhớ thì năm đầu anh học ở Khoa học tự nhiên thì anh cũng không có nhiều buổi train lắm, chắc là được khoảng 4-5 buổi train và nội dung training các đề regional các năm trước do các anh khóa trước sưu tầm. Còn sau này qua NUS thì phần mà được đầu tư nhiều hơn là phần thi chọn team, các thầy soạn đề rất là kỹ và thầy cũng sưu tầm nguồn đề từ các bạn cựu sinh viên qua nhiều năm, và anh thấy có một cái NUS làm khá công phu là thầy Steven có soạn hẳn ra một bảng đánh giá sinh viên trong các team, kiểu bạn đó có thi IOI không, có thi HSGQG không, rating trên các nền tảng CP⁵⁴, cả điểm trong **khóa học Competitive Programming của thầy**⁵⁵. Anh nghĩ đó cũng là một cái để mọi người có cái nhìn khách quan về những thí sinh tham dự, từ đó có cách chọn team cho hợp lý. Từ phía thí sinh, theo tâm lý thì mình sẽ muốn chọn bạn của mình vào team, cũng là điều dễ hiểu. Nhưng điều đó thì không có lợi cho trường vì trường luôn muốn là đưa ba người mạnh nhất vào cùng nhau sẽ tốt nhất. Anh nghĩ đây là một phương pháp hay để mình có được cái góc nhìn khách quan về cái cách mà mọi người chọn team để mọi người không có không có ý kiến việc chọn team thế này thế kia thì nó không phù hợp.

Anh có thể chia sẻ về quá trình luyện tập cho ICPC tại Đại học được không?

Khi anh chuyển qua NUS học năm đầu tiên thì anh đã được thi World Finals rồi. Cái duyên có nó cũng khá là kỳ lạ, tại vì khi anh lập đội trong trường thì anh lập team với hai bạn người Việt. Tụi anh cũng đã có cơ hội thi Regional đầu tiên đi chung với nhau ba đứa. Nhưng mà sau đó tới cái Regional thứ hai, đội mạnh nhất của trường anh có một bạn người Singapore có việc bận nên bạn đó không có đi Regional được, xong rồi anh được thầy thuyết phục chuyển qua đi thi cùng cái team đấy. Team đó thì có một bạn có một anh người Việt và một bạn Indonesia, sau đó anh cùng team đi thi cái Regional Jakarta được TOP 3 cho nên đủ điều kiện thi World Finals. Nói chung là cũng hơi buồn cười vì anh được đi theo suất dự bị, xong team này lại được vào WF. Đợt đó cũng là do kiểu anh chưa quen và thân với team nên là tụi anh không luyện tập với nhau nhiều. Khoảng hai ba tuần trước khi thi, tụi anh mới tổ chức kiểu thi thử mấy contest dài dài với nhau hoặc làm luôn những vòng ICPC ở Regional khác mới được đăng trên mạng. Năm đó team cũng biết là không có cơ hội luyện với nhau nhiều nên cũng không có kỳ vọng kết quả cao nên tụi anh được khoảng hạng 49. Tuy vậy thì team anh giải First solve một bài nên cũng an ủi, thì đó là lần đầu tiên anh thi.



Thật ra anh nghĩ kết quả năm đó như vậy vì anh không có luyện tập nghiêm túc lắm. Sau đó thì anh nghĩ là anh đã đi một lần rồi cho nên anh cũng không có muốn đi hai lần trong hai năm đầu tiên, tại vì anh cảm thấy mình vẫn có thể tiến bộ hơn cho nên là năm thứ ba anh mới quyết định thi tiếp nhưng mà năm thứ ba thì anh không được đi World Finals. Vì vậy cơ hội cuối cùng là qua tới năm thứ tư thì có anh RR (anh Nguyễn Thành Trung) mới chuyển qua trường anh học Thạc sĩ bắt đội với anh, lúc này anh bắt đầu làm bài rất là nhiều. Thật ra cơ duyên cũng là từ tầm khoảng

hè trước khi mà anh vào năm tư thì anh Trung cũng có liên lạc với anh nói về kế hoạch vào NUS. Tụi anh cũng rất nghiêm túc trong việc chọn người thứ ba, ban đầu là cũng định nhắm đến một bạn IOI rồi, nhưng mà cuối cùng vì lý do bất khả kháng là bạn ấy không có qua NUS được thì mới tìm một người khác. Tụi anh quyết định chọn một người có nền toán mạnh là bạn Nguyễn Hùng Tâm (Huy chương Bạc Olympic Toán Quốc tế 2012). Cả mùa hè đó thì anh Trung ngồi chỉ code cho bạn Tâm. Anh nghĩ cái cách chọn người cũng khá là thành công, tại vì bạn học khá là nhanh và cái cách suy nghĩ của bạn ấy nó khác với lại cách suy nghĩ của anh và anh Trung. Điều đó giúp cho tụi anh về những bài thiên về toán hoặc là những bài thuần túy về tư duy, quan sát thì bạn đều làm rất là tốt. Dĩ nhiên là những bài yêu cầu cài đặt thuật toán nặng, hoặc là những thuật toán cổ điển thì anh và anh Trung sẽ là người làm. Đó là phần chọn đồng đội. Về lịch luyện tập thường là anh sẽ dành một tuần ba buổi, sẽ là hai buổi tối

⁵⁴CP: Competitive Programming - Lập trình thi đấu

⁵⁵<https://www.comp.nus.edu.sg/~stevenha/cs3233.html>



trong tuần, làm khoảng 4 tiếng đồng hồ và một ngày cuối tuần thứ 7 thì tụi anh sẽ làm một cái contest 5 tiếng. Đó là train team 3 người, còn những contest thi cá nhân trên mạng thì tụi anh vẫn cố gắng rủ nhau thi. Cứ 13-14 tiếng làm cùng nhau như vậy, nghe là thấy cũng nhiều thật, nhưng anh nghĩ tụi anh vẫn có thể làm tốt hơn. Thời điểm đó, anh với bạn Tâm cũng hơi ham chơi, toàn làm bài xong đòi nghỉ sớm để chơi Dota. Lâu lâu gặp mấy bài khó quá mà nghĩ không ra gì nữa thì bảo anh Trung mình train tới đây thôi mình nghỉ. Cái này nghe cũng hơi hơi trẻ con một chút, nhưng nói chung là cũng chấp nhận thôi tại vì tính ra khi mà tụi anh thi Regional thì kết quả cũng đúng như khả năng của team. Tụi anh vô địch một site ở Jakarta, còn site ở Singapore thì được hạng 3. Còn khi thi World Finals thì tụi anh được hạng 15, hơi tiếc là không được mốc huy chương nhưng mà nó cũng đúng với cái thực lực của đội tụi anh nên cũng không có gì để tiếc nuối về phần đó.

So với việc anh thi ICPC ở năm đầu tiên với một đội ở bên Singapore và năm cuối cùng anh thi với một đội toàn người Việt Nam thì theo anh thấy thì việc trao đổi trong quá trình thi giữa team Singapore với lại team toàn người Việt Nam thì có sự khác nhau không?

Anh nghĩ khác nhau hay không là do nó nằm ở việc em có train với team đủ lâu để hiểu được khả năng của đồng đội. Ví dụ anh với anh Trung, tụi anh quen với nhau lâu và cũng nói chuyện nhiều thì khi mà người này nói ra một hai cái thì người kia sẽ hiểu là ý muốn nói gì, mặc dù cái cách diễn tả của mỗi người có thể không đủ chi tiết, hoặc là mình sẽ phát hiện được là họ bị thiếu những nhận xét nào trong cái suy luận của họ. Còn với những với những bạn mà mình ít train cùng, như anh nói là lúc đó trong team thì anh chỉ là dự bị thôi, do vốn team đó năm trước đã đi World Finals một lần rồi. Vì thời gian tụi anh train với nhau không đủ nhiều nên lúc thi là tụi anh chỉ có chia bài ra rồi mạnh ai nấy làm thôi. Bao giờ đi tới bài khó thì có thể là giải thích cho nhau nghe bài đó rồi cũng tự ngồi nghĩ độc lập với nhau cho bài khó đó chứ cũng không có thực sự kiểu ngồi làm việc cùng nhau. Anh nghĩ đó là vấn đề khi mình không có đủ quen với đồng đội của mình.

Quãng thời gian mà anh gắn bó lập trình thi đấu từ cấp ba đến khi anh lên đại học, anh đã gặp khó khăn hay thất bại nào mà sau này nhìn lại anh nghĩ rằng mình đã có thể làm tốt hơn không?

Đối với bản thân anh, mặc dù có hai lần đi World Finals nhưng có một cái mà anh cảm thấy rất tiếc là cả hai lần đó, anh đều không giải được thêm bài nào trong một tiếng cuối khi đóng bảng rank cả. Cảm giác mà khi tới lúc bắt đầu đọc kết quả rồi nhìn các đội khác mở bảng rank lên, nhìn người ta nhảy hạng trong khi đội mình thì đứng yên là một cái cũng hơi buồn. Còn thời cấp 3, năm đầu tiên mà anh đi TST là năm lớp 11, thì thật ra năm đó anh thật sự hy vọng rất là nhiều. Mặc dù nói về mặt trình độ thì mình cũng thua khá nhiều người nhưng mà mình luôn hy vọng mình sẽ được đi thi quốc tế luôn. Kết quả cuối cùng thì anh không được, nói chung là sau đó anh buồn cũng một thời gian dài và đây cũng là thất bại mà anh tiếc nuối nhất trong quãng thời gian học cấp Ba. Còn năm lớp 12 thì tâm lý anh lại thoải mái hơn, tâm lý là không được thì thôi cũng chả sao, nhưng mà không hiểu sao lại được.

Sau này có một cái sự kiện làm cho anh thực sự quay lại luyện tập là khi anh thi kỳ thi của Google Code Jam, lúc đó anh thi mới tới vòng thứ hai mà anh đã anh đã rút không được áo, một vòng thi hồi đó chỉ giới hạn khoảng 1000. Trước đó thì năm nào anh cũng được áo nên tự nhiên thì không được cái lúc đó anh tự chất vấn là tại sao mình lại kém đến vậy. Từ sự kiện đó anh quyết tâm sẽ phải luyện tập lại, đây cũng chính là lúc mà anh bắt đầu quay lại với lập trình thi đấu, chứ thực ra cái thời mà sau khi anh tốt nghiệp xong đi làm thì anh không có không có luyện tập nữa mà anh thỉnh thoảng anh thi cho vui chứ anh không có dành thời gian luyện tập. Thất bại đợt đó khiến quay lại với lập trình thi đấu và ngoài ra anh cũng một phần tìm lại được niềm vui khi tham gia những kỳ thi đó.



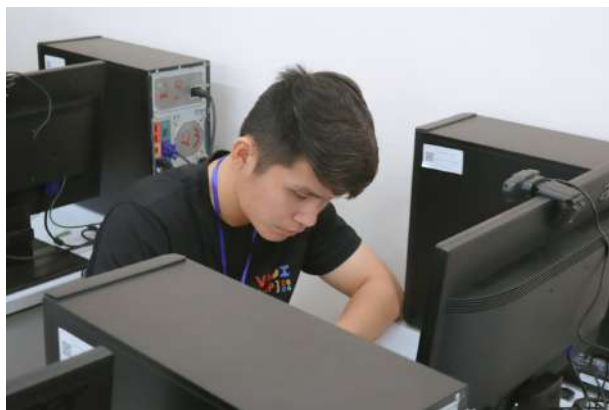


Anh cảm thấy những mục tiêu trong lập trình thi đấu thì có những mục tiêu nào mà anh cảm thấy chưa được hoàn thành không?

Sau khi thi World Finals xong thì anh chỉ có một mục tiêu duy nhất là được đi vòng Finals của một kỳ thi lớn, bây giờ chỉ còn lại có mỗi 2 kỳ thi, một cái của Meta, với lại một cái của AtCoder. Nói chung là hai cái đó đều khó nhưng mà thôi đó là giấc mơ thì mình cứ mơ lớn thôi chứ cũng không có gì (cười).

Được biết sau này cũng có một quãng thời gian mà anh quay trở lại với lập trình thi đấu như anh đã chia sẻ, anh Trung chia sẻ là anh cày còn ghê hơn cả thời xưa anh học đại học thì anh có thể chia sẻ mục tiêu lúc đó của anh là gì không?

Thật ra thì lúc đó anh cũng không có thực sự đặt ra mục tiêu khi mà anh cày, anh chỉ cảm thấy anh làm thì thấy vui. Tại vì cứ mỗi ngày đi làm về xong ăn tối xong là anh bắt đầu ngồi code, anh thấy điều đó nó vui và nó thành một cái thói quen sau này. Như anh cũng chia sẻ, anh chỉ có một mục tiêu vào được một cái Finals, nhưng cái mục tiêu đó thì cực kỳ xa xôi luôn, cho nên là bảo nếu mà để anh cày cho mục tiêu đó thì nó cũng không đúng lắm.



Nhưng mà thật ra anh chỉ muốn là khi giải những bài đó xong mình cảm thấy là nó sẽ có những khoảnh khắc hay có những cái nhận xét rất là hay mà mình tìm ra được thì đó là những cái điều làm anh vui và nó cũng là một cách để mình rèn luyện cái tư duy. Khi anh đi làm thì những bài toán anh giải các thứ ở trên công ty nó là một cái kiểu khác và cũng không có kiểu nặng về tư duy mà sẽ theo kiểu xây dựng hệ thống các thứ, cho nên là làm lập trình thi đấu giúp cho mình làm mới lại cái suy nghĩ của bản thân.

Cơ duyên nào khiến anh chọn NUS là điểm đến tiếp theo trên con đường học đại học của anh sau khi anh chuyển từ Việt Nam qua?

Thực ra ban đầu anh cũng không có ý định đi du học. Nhưng mà sau khi anh được giải quốc tế thì anh bắt đầu xem xét khả năng đi du học của bản thân. Ban đầu anh mong muốn là mình có thể đi Mỹ nhiều hơn nhưng mà năm đó thì hình như học bổng chính phủ đi Mỹ phát sinh vấn đề nên anh không có thông tin về cái đấy, cuối cùng là anh chọn điểm đến du học sẽ là Singapore. Đi Singapore thì nó cũng gần, di chuyển từ Việt Nam thì cũng dễ hơn và chất lượng giáo dục ở Singapore thì khá là tốt. Sau khi anh học xong bốn năm thì anh nghĩ là lựa chọn của mình hoàn toàn đúng đắn. Tại vì cái chương trình học của NUS và tất cả những cơ hội mà anh có được khi học tại NUS thì anh nghĩ là chất lượng tốt so với những địa điểm khác như Úc.

Ngoài những việc học trên lớp ở NUS thì anh còn tham gia những hoạt động ngoại khóa không?

Hồi đó anh chỉ tham gia hoạt động với lại hội sinh viên Việt Nam bên đó thôi. Hầu hết tại anh sẽ sinh hoạt trong những cái sự kiện tổ chức bởi hội sinh viên Việt Nam là chính. Lúc đi học thì anh thường tập trung vào việc học và anh tập trung phần lớn thời gian để làm những cái project (đồ án) trên lớp cùng các bạn. Anh chỉ thích mấy môn làm project các thứ vì nó giúp mình luyện được những kỹ năng làm việc thực tế, gần với những cái đi làm sau này hơn. Còn những môn nặng lý thuyết thì thực ra học cũng vui nhưng anh sẽ không có dành nhiều thời gian lắm. Anh chỉ tập trung vào những môn làm project nhiều hơn. Hồi đó anh học thì nó có hai môn anh khá là thích, một là lập trình Web, môn thứ hai là lập trình



Ipad, lúc đó tụi anh làm một cái game trên Ipad. Khi anh làm những project đó xong thì anh cũng dùng đó làm kinh nghiệm để anh nộp hồ sơ xin thực tập ở bên Singapore luôn.

Ngoài hai môn này thì anh còn có ấn tượng với môn học nào khác tại NUS không?

Ngoài ra thì còn một môn nữa là Optimisation Algorithms (tối ưu thuật toán), thì nó là một môn về thuật toán nhưng ở cấp độ cao hơn. Môn sẽ hơi thiên hướng lý thuyết một chút, nghĩa là sẽ phải làm những thứ liên quan đến chứng minh nhiều hơn. Thay vì chỉ là biết một cái thuật toán thì bây giờ sẽ là chứng minh cận trên cận dưới của kết quả, nói chung có lại một chút với toán.

Trong quá trình học thì anh có thường xuyên trao đổi với các giảng viên, giáo sư nào ở trường không, và anh ấn tượng với thầy/cô nào nhất?

nh thì anh không có trao đổi nhiều với giảng viên nhưng nói về cách dạy thì có người làm anh cảm thấy ấn tượng nhất là thầy Seth Gilbert dạy môn Optimisation Algorithms

Thường thì ở NUS, có thầy Steven Halim, quãng thời gian anh theo học tại NUS thì thầy đã làm coach (huấn luyện viên) đội tuyển ICPC chưa?

À! Có rồi, lúc đó thầy đã bắt đầu làm coach rồi. Thầy cũng dạy tụi anh môn Competitive Programming và sau đó thì anh cũng có làm trợ giảng môn đó cho thầy các khóa sau. Anh nghĩ là điểm đánh giá cao nhất ở thầy Halim là sự nhiệt tình của thầy và sự tâm huyết của thầy dành cho cái phần lập trình thi đấu của trường. Thầy cùng với thầy Tan Sun Teck là hai người đã một mình kéo NUS từ con số không mà đi lên tới cái ngày có được huy chương ở ICPC. Cả những việc như là tổ chức IOI cho Singapore, mặc dù không may mắn là đúng vào cái năm bị dịch thì phải tổ chức online. Nhưng mà anh nghĩ là việc ở một đất nước nhỏ vốn cũng không có nhiều trường học như vậy mà làm được những thứ đó thì nó thể hiện ra cái sự tâm huyết của hai thầy trong cái lĩnh vực này.

Anh có thể chia sẻ về quá trình anh xin học bổng NUS như thế nào và có bí quyết nào để được học bổng của NUS không ạ? Vì theo xu hướng rất là nhiều bạn ở Việt Nam hay vào NUS theo hướng học bổng của Đông Nam Á.

Thật ra theo anh biết là có thể sau này xin học bổng bị khó lên. Thời anh thì các bạn được đi thi quốc tế khả năng được khá là cao. Tại vì hồi đó là anh làm hồ sơ qua trung tâm chuyên làm hồ sơ cho các bạn đi học bên Singapore. Sau đó anh cũng được phổ biến quá trình để nộp hồ sơ đi Singapore như thế nào và nộp hồ sơ xin học bổng ra sao. Sau khi nộp hồ sơ xong thì cũng có 1-2 thầy của NUS bay qua bên thành phố Hồ Chí Minh tổ chức buổi phỏng vấn, anh cũng phải ngồi phỏng vấn nói chuyện với thầy. Nội dung buổi phỏng vấn xoay quanh lý do mình muốn qua Singapore học, rồi sau khi học xong thì mình muốn làm gì, các câu hỏi chủ yếu muốn tìm hiểu về ứng viên. Anh nghĩ là nó cũng không có gì phức tạp lắm, trong quá trình nộp thì phải viết một bài luận ngắn nữa, nói chung là thời của anh nó chỉ có hai cái đó thôi. Sau này anh cũng bắt đầu nghe thấy là có nhiều bạn cũng có huy chương quốc tế nhưng vẫn không được học bổng do dần dần NUS chỉ giới hạn huy chương Vàng thì mới được học bổng, theo anh nghĩ là do quỹ dành cho học bổng đó của bên trường NUS và cả Singapore có thể bắt đầu hơi dần đi rồi thì cái tiêu chí đó sẽ khó hơn, hoặc là bây giờ sau này anh thấy là có nhiều nước khác cũng tham gia cạnh tranh học bổng ngoài Đông Nam Á, mình sẽ phải cạnh tranh với nhiều người hơn cho nên sẽ khó hơn.





Ngoài lập trình thi đấu thì anh có đi làm. Anh có quan tâm đến những công nghệ nào khác không? Và đâu là mảng anh đang quan tâm hiện tại?

Hiện tại anh đang làm về phần Web Application, nói chung anh làm cả front-end lẫn back-end. Phần anh làm cũng không có nhiều cái mới lắm, nhưng mà chủ yếu là những sản phẩm được xây dựng sao cho khách hàng dễ sử dụng. Với anh, anh cảm thấy yêu thích nhất là mặt user experience (trải nghiệm người dùng), thường anh cảm thấy là một số bạn engineer chưa có nhiều kinh nghiệm thì các bạn không để ý nhiều đến phần này, các bạn code một cái tính năng chạy được và không bị lỗi là ổn rồi. Nhưng theo anh nghĩ là cần quan tâm nhất khi xây dựng sản phẩm thì phải làm sao để cho người dùng dễ sử dụng. Tại vì bản thân anh cũng dùng khá là nhiều app rồi, mỗi lần anh dùng app xong thì anh đánh giá cái trải nghiệm của anh nó rất là quan trọng. Như khi dùng một cái app xong rồi vào mà nhìn không biết phải bấm nút gì hết, như anh cần gửi một cái tin nhắn đi mà không tìm thấy chỗ gửi chẳng hạn, đó là những trải nghiệm rất là tệ, và anh thì rất là quan tâm đến cái này. Quan điểm của anh là làm cái app thì luôn làm sao để cho nó càng dễ hiểu cho người dùng càng tốt. Và anh cũng thỉnh thoảng đi đọc một số bài báo về user experience, nghe cũng hơi buồn cười, vì những người học thuật toán như mình thì thường phải thích là những thứ liên quan đến kiểu hệ thống lớn rồi kiểu bài toán mở rộng hệ thống các thứ nhưng mà về riêng anh thì anh thích làm thứ anh cảm thấy mình thích hơn.

Theo anh, lập trình thi đấu so với lập trình khi đi làm thì có cái sự khác nhau như thế nào?

Về coding convention (code chuẩn) và clean code (code sạch) với anh thì hai cái này nó khác nhiều hơn là nó giống, anh nghĩ là tám phần hai cái đó khác nhau, may ra chỉ có hai phần giống nhau thôi. Clean code và coding convention khi đi làm thì thật ra là những cái rất là nhỏ. Tại vì bây giờ khi đi thi thì bản thân anh cũng code sao cho dễ hiểu nhất, sau này khi mình đọc lại code của mình thì mình phải hiểu là mình viết cái gì chứ không thể nào mà mình code xong mình không hiểu thì nó rất là dở. Ngoài ra khi đi làm, thứ nhất là nhiều khi không phải anh viết một cái code mới hoàn toàn mà có khi anh phải vào sửa một cái code có sẵn của đồng nghiệp, thì anh phải làm sao để đảm bảo là cái mà anh sửa không có làm tạo ra bug cho những cái tính năng khác thì đó là một cái khó, đòi hỏi là mình phải có khả năng đọc hiểu code của người khác. Tương tự như vậy thì khi mình viết một cái code mới ra thì mình cũng phải chuẩn bị tâm lý cái code của mình là sẽ có người khác dùng nó và sẽ có người khác làm thêm những cái chức năng dựa trên code của mình, điều tiên quyết phải làm sao để cho cái code của mình dễ hiểu, đủ clean rồi và phải handle được những trường hợp đặc biệt, không xảy ra lỗi. Anh nghĩ đó là những cái quan trọng.



Khi đi làm thì mình sẽ làm trong một cái team rất là lớn, và nhiều khi em sẽ không biết được cái code của em bị người khác sửa. Nếu trường hợp tốt thì người ta sửa xong và nhờ em review code thì cái này không sao. Nhưng mà sẽ có những cái trường hợp mà người ta không nhờ em review code và người ta nhờ người khác chẳng hạn và nhiều khi người đó cũng không biết code đó là gì và sau đó tự nhiên một ngày đẹp trời thì mình phát hiện ra là có một cái bug do người khác sửa code của mình do người đó không hiểu rõ code của mình thì lúc đó mình sẽ tự trách mình là viết code không được tốt, không có đủ để hiểu cho người ta thôi. Đây là một cái ví dụ về cái sự khác nhau giữa hai cái, code đi làm là mình phải xác định nó sẽ tồn tại trong một thời gian dài, có thể 1-2 tháng hay 1-2 năm, phải làm sao trong quá trình đó cái code đó không phát sinh vấn đề, còn code lập trình thi đấu thì em chỉ code để nộp bài và sau một hai tiếng là đã bỏ đi rồi, không



còn dùng nó nữa thì cái việc code không clean lắm cũng không ảnh hưởng. Ngoài ra, khi đi làm thì mình còn phải viết test cho chính code mình viết ra, nên đó là những cái khác biệt chính mà mình sẽ dễ nhận ra khi mình đi làm so với khi mình code CP.

Đối với anh thì những kỹ năng nào trong lập trình thi đấu mà có thể giúp người học lập trình thi đấu có lợi thế hơn khi học những kiến thức cho việc đi làm?

Đầu tiên trước khi đi làm thì mọi người phải đi phỏng vấn và anh nghĩ là cái duy nhất là khi học lập trình thi đấu đem lại lợi thế đó là cơ hội vượt qua được những vòng phỏng vấn cao hơn. Vì bây giờ phần phỏng vấn cũng sẽ hay xuất hiện những câu hỏi liên quan đến lập trình thi đấu như về thuật toán. Các bạn học lập trình thi đấu thì các bạn sẽ có một cái nền thuật toán khá là vững thì các bạn sẽ hiểu được những cái thuật toán cơ bản như hash, BFS/DFS, ... Khi tiếp xúc một bài toán thì sẽ dễ hình dung được áp dụng thuật toán nào, cũng như các giới hạn cho bài toán đó, và đánh giá được những cách tiếp cận. Tại vì thật ra là khi đi làm thì những cái thuật toán đó người ta đã code hết cho rồi nên mình chỉ cần đọc hiểu những thư viện đó, xem nó làm cái gì rồi dùng nó thôi chứ ít khi nào mình sẽ phải viết lại từ đầu. Vấn đề bây giờ là mình phải có kiến thức nền vững thì mình mới đánh giá được những cái mà người ta viết đó thì người ta dùng cái gì để viết và mình biết nó có phù hợp với bài toán mình hay giải không. Anh nghĩ đó là một cái khá là quan trọng. Ngoài kỹ năng viết code còn có kỹ năng debug, đây cũng là một cái kỹ năng mình cần có khi đi làm, tại vì có những cái những cái bug khá là sâu và đi qua nhiều lớp, đôi khi mình sẽ cần một sự tập trung và khả năng đọc code, đó cũng là một trong những cái mà anh nghĩ những bạn học lập trình thi đấu thì có thể là sẽ làm tốt. Vì các bạn đọc những cái thuật toán mà các bạn viết trong lúc thi nó cũng đã rất là phức tạp rồi, nếu các bạn debug được trong lúc thi thì khi các bạn đi làm cũng khả năng cao là bạn có thể tìm ra được những cái bug nó nhanh hơn.

Ngoài thời gian anh dành cho lập trình thì anh còn có sở thích hay là hoạt động nào khác để giải trí không? Như chia sẻ là anh có chơi game DOTA nhưng em có xem thì anh đã dừng chơi Dota từ năm 2019 rồi?

Anh giải nghệ cũng lâu rồi. Nói chung là ngày xưa, thời điểm trước khi anh nghỉ chơi game thì ngoài Dota anh cũng có chơi FIFA nữa. Anh chơi kiểu rất là try hard (chăm chỉ), có những tuần anh chơi FIFA tới 40-50 tiếng, có những hôm là ngồi 8-10 tiếng để anh leo rank do anh cũng là một người có tính thi đấu hơi nhiều so với người khác. Sau này anh không chơi game nữa thì anh cũng có nhiều sở thích khác. Bây giờ anh thường dành thời gian xem phim, xem TV hoặc là đi du lịch. Anh với lại vợ anh thì tụi anh cũng đi du lịch khá là nhiều và cũng cố gắng đi những nơi khác nhau để mình tìm hiểu những cái nền văn hóa và trải nghiệm những cái những thứ mới. Mỗi lần đi thì tụi anh cũng quay lại những trải nghiệm đó, giờ mỗi lần đi về là sẽ phải ngồi làm video editor để làm những clip đấy. Còn thể thao thì anh cũng có chơi đá bóng, chơi với công ty anh, bình thường anh cũng hay xem bóng đá. Anh nghĩ là thật ra khi mình qua cái thời sinh viên rồi thì cái quỹ thời gian của mình nó cũng không đến mức là không có nhưng mình sẽ phải tìm cách để mình chọn những cái sở thích nào mình muốn dành thời gian cho nó thì khi đó thì mình sẽ tìm được cái điểm cân bằng trong cuộc sống của mình.

Như mọi người cũng biết thì anh Nguyễn có một channel youtube nhỏ, mọi người có thể tìm xem các video du lịch của anh ấy tại [đây](https://www.youtube.com/@binbendumonde) ⁵⁶.

Vâng! Vậy anh có chia sẻ là anh có chơi đá bóng và anh cũng có xem đá bóng luôn thì thường thì giải đấu anh xem sẽ là giải đấu nào và anh là fan của câu lạc bộ nào?

Anh xem Ngoại Hạng Anh, cũng giống như nhiều người Việt Nam khác, anh là fan của Arsenal.

⁵⁶<https://www.youtube.com/@binbendumonde>



Đề kết thúc phần phỏng vấn, anh có lời nào muốn nhắn nhủ đến các bạn độc giả của tạp chí, hay lời khuyên cho các bạn trong việc học tập, lập trình thi đấu cho kỳ thi HSGQG, hay sắp đến là kỳ thi TST không ạ?

Anh cũng có biết là kỳ thi HSGQG vừa qua, anh nghĩ là thời điểm ra mắt tạp chí thì mọi người cũng đã biết được kết quả. Anh hy vọng các bạn đều được giải như mong đợi và sẽ không bị mắc những lỗi khiến mình mất điểm không như mong muốn thì đó là điều tốt nhất. Những bạn xứng đáng thì sẽ được vào TST, anh cũng hy vọng là tới kỳ thi TST, các bạn được thi với một cái format mà nó gần với IOI hơn thì các bạn sẽ có thể thể hiện hết khả năng của mình và đạt được những kết quả tốt. Còn với những bạn mà không may mắn, là không được vào TST mà chỉ dừng lại ở vòng HSGQG thì nếu mà các bạn lớp 10, lớp 11 thì cơ hội vẫn còn rộng mở cho các bạn còn nếu mà các bạn lớp 12 thì vẫn còn kỳ thi ICPC ở đại học. Anh nhận thấy là có rất là nhiều người lên đại học người ta mới giỏi lên rất là nhiều, ví dụ như thời anh có anh Lăng Trung Hiếu, sau đó thì có chemthan (Nguyễn Ngọc Trung), hoặc gần đây có bạn Lê Phước Định, hay bạn Đỗ Đình Đắc nữa, thì đó là những con người mà cấp 3 học Toán và lên đại học mới học Tin. Những bạn mà đã có nền Tin từ cấp 3 rồi, thì nếu có niềm đam mê với Tin học thì tiếp tục theo đuổi, cơ hội ở đại học vẫn còn rất là nhiều. Và kể cả nếu ở đại học tụi em không có làm được thì sau khi tốt nghiệp đại học, nếu em thật sự đam mê thì em vẫn sẽ làm được thôi. Tại vì không chỉ là anh mà anh nhìn thấy là trên thế giới cũng có rất nhiều người, một vài ví dụ thì có những bạn bắt đầu lập trình từ năm 26-27 tuổi nhưng mà sau đó người ta vẫn học và người ta vẫn tham gia được những kỳ thi Finals của những kỳ thi lớn trên thế giới. Anh nghĩ cái quan trọng mình cảm thấy lập trình thi đấu là đam mê và mình muốn theo đuổi nó. Và mình cảm thấy thích điều đó thì mình cứ làm thôi! Còn với các bạn mà cảm thấy đạt được mục tiêu của mình rồi, mình muốn đi tìm thử thách khác thì đó cũng hoàn toàn là hợp lý. Tại vì lập trình thi đấu là một phần chứ không phải là tất cả cuộc đời của tụi em cho nên là cũng không cần phải cố gắng mà đi theo nó cho tới hết.

Cảm ơn phần chia sẻ vô cùng tâm huyết từ anh, Ban biên tập Tạp chí VNOI 2025 nói chung cũng rất là cảm ơn anh đã nhận lời phỏng vấn và tham gia buổi phỏng vấn ngày hôm nay. Chúc anh sắp tới sẽ có nhiều sức khỏe và đạt được những cái mục tiêu mà mình đã đề ra ạ.

Cảm ơn em. Chúc em và ban biên tập luôn có nhiều thành công và năm mới mọi người luôn giữ vững tinh thần và ra được một tạp chí Tết hay để cho cộng đồng có thể có một thức ăn tinh thần trong dịp Tết.



Phỏng vấn bác Nguyễn Long

Interviewer: Vương Hoàng Long – ICPC World Finalist 2021



Cảm ơn bác dành thời gian phỏng vấn cho tạp chí VNOI - xin giới thiệu đến bác Nguyễn Long - Tổng thư ký của Hội Tin học Việt Nam, là người có đóng góp lớn nhất cho nền Tin học Việt Nam trong vòng 10 năm qua. Rất là cảm ơn bác đã nhận lời phỏng vấn. Để bắt đầu cuộc trò chuyện thì cháu muốn bác kể về lịch sử hình thành của nền Tin học Việt Nam, từ những ngày đầu tiên cho đến hiện tại đã trải qua những thăng trầm nào?

Lịch sử phát triển Tin học Việt Nam

Để đi vào chi tiết lịch sử thì nó từ rất lâu rồi. Tại vì Việt Nam, thời ngày xưa ngày xưa, xuất phát những người đầu tiên theo đuổi Tin học là những người giỏi Toán, sau đây thì họ 'nhập cuộc' rất là nhanh với xu hướng thế giới, minh chứng là sau khi có Viện Toán học Việt Nam, lúc còn chưa có chữ Tin học thì thành lập ra gọi là Viện Khoa học Tính toán và Điều khiển. Nếu mà nói CNTT, Tin học Việt Nam thì ngày xưa ngày xưa các máy vi tính thì nước mình không có, hay có rất ít, cả nước có vài ba cái, chả có ý nghĩa gì, đặc biệt là những năm 80 khi mà thế giới bắt đầu chế tạo máy tính, tạo thành trào lưu. Trào lưu này bắt đầu đồng thời với sự phát triển về khái niệm Tin học, còn CNTT Việt Nam thì bắt đầu đồng thời với sự xuất hiện của những chiếc máy tính Apple. Lịch sử chính thức được viết là khi nhà nước tổ chức nêu ra vấn đề vào năm 1962, nhưng thực chất bắt đầu có sự tiến triển đầu tiên là đầu những năm 80, và khi đến giữa năm 80 thì Việt Nam bắt nhịp rất nhanh với chuyện máy vi tính vào Việt Nam từ hãng Apple, rồi dần lên đến máy IBM..., những chiếc PC đầu tiên, và và việc mình bắt nhịp như vậy chỉ chậm sau khối Châu Á - Thái Bình Dương một tí thôi! Và đó là lịch sử.

Tính ra Việt Nam nhập cuộc Tin học như thế là cũng rất là nhanh rồi, chiến tranh thống nhất là năm 1975, mà nhập cuộc đến năm 1982, 1982 là có chiếc máy tính đầu tiên rồi

Năm 1982 lúc đấy còn rất kì lạ là Việt Nam còn chưa đổi mới, 1983 mới đổi mới, nhưng mà 1982 lúc đấy có những tổ chức của Mỹ đứng ra giúp Việt Nam, tôi nhớ lúc đấy là một tổ chức quốc tế tặng Viện Hàn Lâm Khoa học Việt Nam hẳn 40 'quả táo' (máy tính Apple) đời đầu vào năm đó. Việt Nam cũng được nhờ những người mà khởi đầu có tầm nhìn rất tốt như là bác [Phan Đình Diệu](#) ⁵⁷.



GS Phan Đình Diệu và các nghiên cứu sinh, thực tập sinh tại Pháp

chuyện đấy rồi, còn để mà nói về chiếc máy tính đầu tiên do ta tự chế tạo từ những năm 80, khi mà mình có cơ hội nhưng mà mình không nắm được cơ hội thì rất là khó thành công. Nhưng cũng kể từ đấy mà bắt đầu phong trào Tin học. Những người làm Tin học hồi đấy thì rất đa dạng, kể cả học gỗ được văn bản để kiếm tiền cũng là 1 nghề. Còn đối với những người từ Toán chuyển sang thì mới bắt đầu có khái niệm lập trình. Bản chất số người học lập trình rất ít, bởi vì nếu muốn lập trình thì phải xếp hàng để



VT80 – chiếc máy vi tính đầu tiên của Việt Nam

Bác có tầm nhìn rất tốt và quan hệ tốt với các nước, bác đưa được những lứa thanh niên đầu tiên của Việt Nam sang các nước khác du học, bắt đầu nghiên cứu về vi mạch và các bộ vi xử lý như chip Intel 8085 các thứ. Nhưng cũng có một số người tỏ ra tiếc nuối là Việt Nam hồi đấy mà Mỹ đầu tư được thì Việt Nam có thể là nước đầu tiên thế này thế kia (cười). Vừa rồi như là có trường hợp như là anh Chính (Nguyễn Trung Chính) ở công ty CMC cũng có quan điểm như thế. Thật ra hồi đấy Việt Nam bị cấm vận, nên nên để tự chủ và lắp ráp được chiếc máy tính rất là khó. Hồi đấy có khởi động dự án dây chuyền lắp ráp máy tính PC đầu tiên thì có nhóm CMC đảm nhiệm chuyện đấy, và sau này tự nước ta cũng thực hiện được

⁵⁷https://vi.wikipedia.org/wiki/Phan_%C4%90%C3%ACnh_Di%E1%BB%87u



chạy chương trình trên máy tính lớn, có khi xếp hàng mấy ngày thì nó mới chạy ra nó báo lỗi thì về làm lại. Hồi đấy bảo những người giỏi có thích lập trình hay không thì cũng chưa chắc, tại vì nếu không được tiếp cận trực tiếp lập trình thì nhiều người giỏi chắc người ta cũng không muốn học. Và cái ngôn ngữ lập trình ban đầu nó cũng rất là sơ khai. Lúc trước học là mã máy, hoặc là nếu chạy máy lớn thì học PL/I hoặc Fortran theo một cái format rất là chặt chẽ mà chỉ thiên giải bài toán kĩ thuật, thế thì hầu hết là những người theo học mấy cái môn kĩ thuật hay cần giải một số các bài toán quy hoạch, bài toán tuyến tính, .. thì mới theo, chứ còn lại không có ai theo. Sau đó, ngôn ngữ đầu tiên mà đưa về Việt Nam dưới dạng có một quan hệ người - máy mà nó gần gũi như bây giờ xuất hiện trên máy 'quả táo 2' (Apple 2) là ngôn ngữ Basic.

Sau Basic thì bắt đầu xuất hiện ra rất nhiều thứ khác, nó là sự đột phá của IBM PC. Ngay từ đầu, hệ điều hành của máy Apple thuần túy là chỉ dành cho các máy của Apple thôi, cũng giống như bây giờ, Safari hay là các ứng dụng khác. Sau đó Microsoft xuất hiện, kéo theo là sự phổ biến của các hệ điều hành, các ngôn ngữ lập trình cũng từ đó lên ngôi. Nếu không tính các ngôn ngữ lập trình cổ điển như PL/I, Fortran vẫn còn tồn tại thì đó là trào lưu của ngôn ngữ lập trình Pascal, tất cả mọi thứ sau này đều bắt đầu từ Pascal. Nhưng sau đó thì người lập trình cảm thấy Pascal không còn phù hợp nữa nên bắt đầu xuất hiện thêm những thế



Máy tính Apple 2

hệ ngôn ngữ mới như C, C#, C++, Java... Bắt đầu từ đấy thì nhà nước mới chính thức thành lập ra 5 khoa CNTT đầu tiên trên cả nước ở các trường Đại học vào cuối năm 1980, trong đó có trường Đại học Tổng hợp Hà Nội, Đại học Bách Khoa Hà Nội, Học viện Kỹ thuật Quân sự, Đại học Tổng hợp TP.HCM mà sau này là trường Đại học Khoa học tự nhiên TP.HCM, có mấy trường thôi à!

Đấy là lịch sử của CNTT tại Việt Nam. Vậy theo bác nền móng Tin học lập trình thì đâu là bắt đầu từ những năm bao nhiêu ạ?

Sau khi Việt Nam thì các môn tự nhiên như Toán, Lý, Hóa như bình thường rồi. Thật sự năm 1975, các nước khác đã đi thi lần thứ 16 thì mình mới gia nhập, khi tôi đi thi là lần thứ 17. Ngày xưa IMO không gọi là IMO như bây giờ đâu, mà gọi là MOM theo tiếng Nga chung khối XHCN. Sau này thì Việt Nam mới tích cực gia nhập chứ hồi vừa chiến tranh thì hầu hết các nước XHCN đang cai, Việt Nam xin được chỗ nào tài trợ tiền máy bay thì mới thi, còn không thì không đi thi. Sau này sướng hơn, khoảng năm 1982-1987 thì bắt đầu thì bắt đầu xuất hiện lứa đi thi Tin học đầu tiên.

Lịch sử Olympic Tin học Sinh viên Việt Nam

Lần đầu tiên mình đi thi IOI⁵⁸ là năm nào ạ?

Cái đấy phải xem lại lịch của khóa thi IOI đầu tiên (cười). Nhưng Việt Nam là một trong những nước góp mặt tại kỳ IOI đầu tiên, tôi chỉ nhớ Việt Nam lần đầu đi thi Olympic Tin học Quốc tế thì có nhờ Hội Tin học cử người - một trong những người được cử đi khoá đầu tiên chính là thầy [Nguyễn Việt Hà](#)⁵⁹, về sau là Hiệu trưởng Đại học Công nghệ, ĐHQG-HN, thì thầy là khoá đầu tiên mà không phải từ trường chuyên đại học cử đi mà là do Hội Tin học tiền cử. Một trong những người đầu tiên dẫn đội đi thi và từng ngồi trong hội đồng giám khảo ra đề Việt Nam là thầy Nguyễn Việt Hà.

Vậy theo bác thì các kỳ thi Tin học trong nước lần đầu tiên chúng ta bắt đầu tổ chức thi để tìm kiếm tài năng thì là bao giờ ạ?

Kỳ thi đầu tiên trong nước để mà kể lại thì là cuối những năm 80 thì các khoa CNTT ở các trường đại học bắt đầu tuyển sinh ngành CNTT, ngành Tin học. Hồi đấy thì các khoa vẫn đang lẫn lộn giữa chữ Tin học và CNTT, rồi 5 trường có khoa CNTT đấy bắt đầu huy động sinh viên để thi lập trình. Đến năm 1992 hay 1993 thì có kỳ Olympic Tin học Việt Nam đầu tiên.

Năm 1993 ạ? Tức là năm 2023 là đã được 30 năm rồi?

⁵⁸IOI: Kỳ thi Olympic Tin học Quốc tế.

⁵⁹https://vi.wikipedia.org/wiki/Nguy%E1%BB%85n_Vi%E1%BB%87t_H%C3%A0



Nếu mà tính đến năm nay là năm 2025 là năm thứ 34, sắp sửa 35 năm rồi. Năm 2022, tổ chức Olympic ở Đại học FPT làm kỉ niệm 30 năm OLP đấy! Hồi đấy có những thầy rất nổi tiếng trong hội đồng giáo sư ngành như [Phạm Thế Long](#)⁶⁰, [Nguyễn Thanh Thủy](#)⁶¹, thầy Hồ Sỹ Đàm, ... tham gia tổ chức bắt đầu từ phong trào Đoàn thanh niên.



Hội đồng giám khảo OLP từ 1997 - 1999

Vậy từ những năm 2000 thì kỳ thi mới được Hội Tin học tổ chức hay sao ạ?

Không, Hội Tin học vẫn là một trong những đơn vị tổ chức ngay từ đầu nhưng mà vẫn làm dựa theo phong trào cũ của các trường và được Bộ Giáo dục và Đào tạo ủng hộ, nhưng không hiểu tại sao hồi đấy người ta lại nghĩ thi Olympic lại chỉ dành cho những người học CNTT nhưng mà chưa được giải!

Đến năm 1999-2000, có 1 điều đặc biệt nữa là trong hệ thống đại học có kỳ thi ICPC⁶⁵ do Hiệp hội Máy tính (ACM) nghĩ ra. Hầu như chương trình đại học CNTT bây giờ là theo chuẩn ACM hết. Lúc bấy giờ so với các nước lớn thì lịch sử của họ cũng 40-50 năm tham gia rồi, các thầy ở Việt Nam lúc đó chưa dám tham gia ngay mà phát triển một phần thi riêng trong kỳ thi Olympic.

Bây giờ, tôi cũng không biết họ dịch làm sao nhưng tạo ra bộ môn thi tập thể gọi là [Lều chõng](#)⁶⁶ (Trong quy chế của kỳ thi Olympic Tin học Sinh viên Việt Nam vẫn đề cập tới cái tên này như tôn vinh và nhắc lại giá trị lịch sử của kỳ thi). Lều chõng chính là ICPC!

Đại khái thế! Tại vì hồi đó ở Việt Nam thì Internet năm 1997 mới bắt đầu vào, nên các thầy cũng không có nhiều thông tin về kỳ thi trước đó. Đề thì dễ rồi, mà không biết người ta chấm như thế nào! Hồi đó chấm tự động như bây giờ là rất hiếm. Mình nộp bài qua USB xong mấy thầy chấm theo kiểu là IOI (Hình thức chấm IOI khi chưa được chuẩn hóa bằng máy chấm thì hội đồng giám khảo phải chấm code bằng cách đọc code và tự biên dịch ra ý tưởng). Các thầy giám khảo hồi đấy toàn xuất phát từ dẫn đoàn hay thi IOI ra thì cứ chấm theo IOI thôi (cười).

Quay lại kỳ thi vào khoảng năm 2000, rất nhiều thầy có tiếng trong ngành bây giờ nổi danh từ hồi đấy, như thầy [Trần Minh Triết](#)⁶⁷ của trường Đại học Khoa học tự nhiên, TP.HCM bây giờ cũng đã từng

Ban đầu chỉ có 5 trường thi với nhau như thế rồi CNTT bắt đầu có người học tăng dần nên thành một trào lưu. Sau 5 năm tổ chức (năm 1997) thì kỳ thi bắt đầu có quy mô hoành tráng hơn, năm đấy là được tổ chức ở Quy Nhơn, lúc đấy trên cả nước có độ khoảng mười mấy trường có khoa CNTT dự thi và kỳ thi cũng chứng kiến nhiều tài năng trẻ từng đạt giải IOI tham gia. Nổi bật trong lứa đấy thì có thầy [Phạm Bảo Sơn](#)⁶², [Trần Quốc Long](#)⁶³. Tuy vậy, sau năm 1997 thì Bộ Giáo dục và Đào tạo lại cấm các bạn đã được giải HSGQG⁶⁴ không được thi Olympic làm cho cuộc thi kém vui! Khoảng sau đấy thì kỳ thi cũng không có gì đặc biệt thêm.



Lều chõng tại OLP năm 2006 - Đại học Bách khoa Hà Nội

⁶⁰<http://hdgsnn.gov.vn/files/anhbaiviet/files/2020/LLKH%20Thanh%20vien%20cac%20HD%20nganh%202020/4-HD%20CNTT%20LLKH%202020/4-HD%20CNTT-GS%20PT%20Long.pdf>

⁶¹<https://cmcu.edu.vn/giang-vien/nguyen-thanh-thuy/>

⁶²https://vi.wikipedia.org/wiki/Ph%E1%BA%A1m_B%E1%BA%A3o_S%C6%A1n

⁶³<https://scholar.google.com/citations?user=tnDavuYAAAAJ&hl=en>

⁶⁴HSGQG: là viết tắt của kỳ thi Học sinh giỏi Quốc gia.

⁶⁵ICPC: International Collegiate Programming Contest - Cuộc thi Lập trình Quốc tế lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu.

⁶⁶<https://www.olp.vn/olympic-tin-h%E1%BB%8Dc-sinh-vi%C3%AAn/quy-ch%E1%BA%BF-n%E1%BB%99i-quy>

⁶⁷<https://scholar.google.com/citations?user=lt2ATkkAAAAJ&hl=en>



đạt giải Nhất, Nhì khoảng độ thời gian đó. Kỳ thi Olympic bắt đầu 'đổi màu' khoảng năm 2004. Thời điểm này tôi mới nghĩ rằng là tại sao kỳ thi lại không cho phép những người giỏi tham gia, điều đó tôi thấy không chấp nhận được! Vì vậy sau năm 2003 tôi có đề xuất và đến năm 2004 thì kỳ thi có lập ra một khối riêng cho các bạn ấy gọi là Siêu Cúp, những ai giỏi nhất thì thi bên này.



Sinh viên Lâm Xuân Nhật vô địch Siêu CUP OLP 2004

Bước đầu tiên để đưa Việt Nam hội nhập với thế giới là tôi phải liên hệ ông Hwang (chủ tịch ICPC Châu Á), về ông Hwang thì ông là người có công phổ cập ICPC ra cho toàn bộ châu Á, một người rất có công, tuy ông ý cũng có hơi lạ thường (cười). Ngay sau khi kết nối thì ông Hwang bay sang Hà Nội luôn, sau khi đồng ý hội nhập thì tôi cũng nhận lời Việt Nam sẽ đăng cai một Regional đầu tiên vào năm 2006. Về số suất để thi World Finals cho Regional Hà Nội năm đó thì tôi cũng chốt là sẽ chọn 3 đội tuyển xuất sắc nhất và bên ICPC họ cũng hỗ trợ trong việc tổ chức tạo cơ hội các đội Việt Nam lấy suất thi World Finals.

Những năm đầu thì Trường Đại học Công nghệ có dắt đội qua Regional Ấn Độ để tranh suất World Finals và nghĩ là 'ăn' được các đội bên đấy, xong rồi qua đấy thi kiểu gì 'out' luôn. Trường Đại học Khoa học tự nhiên cũng đi thi ở Regional đầu đấy nhưng cũng không được. Duy chỉ có Đại học Bách khoa TP.HCM là liều đi sang Regional Iran, sang Iran đầu thì lại đứng được rank 4.

Vậy Bách khoa TP.HCM là team đầu tiên được đi WF ư?

Đúng rồi, năm 2006 Bách khoa TP.HCM là đội đầu tiên được đi World Finals ở Việt Nam, đây là đội của Phạm Hữu Ngôn. Sau đó đến tháng 4 năm 2006 thì Việt Nam có lần đầu tiên tổ chức ICPC Regional ở Đại học Công nghệ Hà Nội. Phần còn lại là lịch sử.

Các năm sau đó thì Việt Nam đều có đội góp mặt tại ICPC World Finals. Ở thời gian đầu tiên thì thế giới họ không tin Việt Nam đâu. Họ vẫn đánh giá là Thái Lan nhỉnh hơn Việt Nam mình. Những năm 2007 - 2009, đoàn Việt Nam đi sang Regional Thái Chulalongkorn thi đông lắm, có những lần mười mấy đội đi thi và sang đấy thì Việt Nam 'xoi' hết giải của họ. Vậy nên chúng ta là những học sinh/sinh viên Việt Nam đã khẳng định vị trí của mình trên thế giới rất là lớn. Việt Nam gần như liên tục mấy năm là

Năm đầu tiên có Siêu Cúp thì xuất hiện Lâm Xuân Nhật (CTO Autonomous.ai) từng được Huy chương Đồng IOI vô địch Siêu Cúp lần đầu tiên. Năm đấy cũng chứng kiến màn ra mắt của các bạn học sinh giỏi rất xuất sắc mà đã nổi danh từ lâu như Phạm Hữu Ngôn (CEO AhaMove), Lê Thụy Anh (Huy chương Đồng IOI 1995), ... tham gia thi. Còn về ICPC, sau khi các bạn đã từng thi IOI hay Olympic đi học và quay về như [Bùi Thế Duy](#)⁶⁸, Phạm Bảo Sơn, [Lê Sỹ Vinh](#)⁶⁹ học ở Úc hay Châu Âu, học xong quay về thì có nói với tôi là bên ấy học và để chiến đấu đại diện trường đi thi World Finals⁷⁰ là rất khó và đặt vấn đề với tôi là sao nước mình không sớm hội nhập để có một đội tuyển đại diện chính Việt Nam đi thi. Tôi nghe xong thì cũng đồng ý luôn.



CJ Hwang - Chủ tịch ICPC Châu Á

⁶⁸https://vi.wikipedia.org/wiki/B%C3%B9i_Th%E1%BA%BF_Duy

⁶⁹<https://scholar.google.com/citations?user=40kW2KsAAAAJ&hl=en>

⁷⁰World Finals: Cuộc thi ICPC cuối cùng trong một mùa giải bao gồm những đội xuất sắc nhất của các trường đại học trên thế giới vượt qua vòng loại vùng (Regional contest)



năm nào cũng có ít nhất một đội vào World Finals.



Đoàn Vietnam đầu tiên tại ICPC World Finals 2006

đều có cơ hội đi World Finals.

Những năm đầu tiên Việt Nam tổ chức Regional đây thì các đội Trung Quốc nhào sang rất đông, các đội Bắc Đại, Thanh Hoa đủ hết, năm ấy là tất cả 80 đội chỉ nhìn mỗi đội Thanh Hoa thi thôi, đội đây có một bạn nữ code chính, vô địch luôn!

Việt Nam bắt đầu tổ chức Regional đều đặn là từ năm bao nhiêu ạ?

Thật ra có mỗi 2007 mình không tổ chức, sau đó năm 2008 thì mình có tổ chức lại, sau này Thái Lan họ bỏ tổ chức Regional dần thì mình làm đều hơn. Mãi sau Thái Lan bỏ hẳn thì mình làm nốt phần họ, liên tục 7, 8 năm như vậy.

Năm 2007 thì cũng có một chuyện kỳ lạ, Việt Nam năm đấy có một đội toàn sinh viên năm nhất rất mạnh và hoành tráng, tôi gọi đấy là đội 3K gồm [Lê Đôn Khuê](#)⁷¹, [Khúc Anh Tuấn](#)⁷², [Hứa Khắc Nam](#)⁷³, ban đầu ai cũng tưởng đội đấy chắc một suất đi World Finals, thế mà lại có một đội của một anh năm trên là [Lê Huy Bình](#)⁷⁴, đội đấy có mỗi Bình code thôi, một bạn thì dịch đề Tiếng Anh, một bạn thì đi theo đoàn. Đội 3K đấy về sau tản đi, bạn thì sang NUS (National University of Singapore), bạn thì sang NTU (Nanyang Technological University) nhưng mỗi người thì cũng



Các đội tuyển VN tại ICPC Asia Phuket năm 2009

Theo bác thì các lần tổ chức đến giờ bên mình có gặp khó khăn nào không?



Hội đồng giám khảo OLP-ICPC năm 2006

Thật ra bên mình cũng rất là may một cái là được các thầy cũng như các anh đi đầu ấy như Bùi Thế Duy, Phạm Bảo Sơn, Lê Sỹ Vinh, hay là thầy Việt Hà thì họ đều là những người đã từng tham gia các kỳ thi rồi. Các thầy ấy hỗ trợ kỳ thi rất là tốt, cộng với một lứa các bạn học ở Singapore về hỗ trợ, đặc biệt là các bạn từ NTU, chứ không phải các bạn từ NUS như bây giờ. Thời điểm đấy NTU rất là mạnh, họ có quay về giúp. Còn việc ra đề thì ngay khi xuất hiện Siêu Cúp là phía tổ chức cũng phải chuẩn bị một tổ ra đề sẵn sàng rồi, vì đề Siêu Cúp không phải đề như bình thường nữa, mà là đề cho các bạn rất là giỏi thi đấu với nhau, cho nên phía tổ chức gần như không gặp khó khăn trong việc ra đề. Cái vấn đề lớn nhất bên mình gặp phải hồi đó chỉ còn là vấn đề kỹ thuật, mạng máy tính này nọ thôi. Hồi đấy Việt Nam còn nghèo nên các bên hỗ trợ phần cứng tìm kiếm rất khó, mạng máy tính thì chưa được mạnh như giờ, hệ thống thi thì hồi đấy do bên ACM họ làm chủ nên rất phập phù và không ổn định. Sau này nhận thấy là mình phải tự làm chủ cái hệ thống thi đấy, chứ nhiều khi tổ chức thi mà cũng hên xui.

Nếu có gặp vấn đề thì chắc chủ yếu là gặp bên ICPC, chứ bên Olympic Sinh viên thì vô tư! Nếu chẳng may đang thi mà máy sập giữa chừng thì chỉ cần tắt khởi động lại máy chạy tiếp thì các bạn vẫn thi như

⁷¹https://cphof.org/profile/topcoder:beo_chay_so

⁷²<https://cphof.org/profile/topcoder:ktuan>

⁷³<https://cphof.org/profile/icpc:Khac%20Nam%20Hua>

⁷⁴<https://cphof.org/profile/icpc:Huy%20Binh%20Le>



thường. Trong ngần ấy năm thi cũng may là chưa có năm nào xảy ra trục trặc cả.

Theo thời gian thi Việt Nam đã tiến bộ lên như nào ạ?

Tổ chức thì bắt buộc phải tiến bộ thôi. Do nhận biết được Việt Nam hội nhập với thế giới sau nên về đào tạo và đẳng cấp mình đi sau thế giới rất nhiều, vậy nên những năm đầu ngay cả những bạn giỏi nhất được đi World Finals thì trong khoảng 6-7 năm đầu tiên đó các đội Việt Nam cũng chỉ làm được khoảng 1-2 bài thôi, có những năm có đội không giải được bài nào. Để dễ hình dung thì các bạn cứ nhìn hình ảnh của Thái Lan đi thi bây giờ thế nào thì chắc là cũng như Việt Nam mình hồi xưa. Sau 7 năm thì mình cứ tiến bộ lên dần, về công tác ra đề thì cũng rất may mắn là có những bạn ngày xưa đại diện World Finals cũng quay về và giúp công tác rất nhiều, điển hình như là Lê Đôn Khuê, điều này giúp cho chất lượng đề của kỳ thi ICPC tại Việt Nam cũng dần tăng lên.

Đến sau năm 2010, việc có đội Việt Nam đi thi World Finals mà trên 4 bài mới bắt đầu xuất hiện. Năm đỉnh cao nhất của một đội toàn người Việt Nam trước đội của [Bùi Hồng Đức](#)⁷⁵ thì có năm của Nguyễn Tấn Sỹ Nguyên và Nguyễn Thành Trung tiệm cận đạt được huy chương. Trong khi đó thì đội của NTU cũng bắt đầu mạnh lên, hồi đấy được đánh giá là mạnh hơn Việt Nam, điển hình như khóa của Nguyễn Thành Trung, [Phạm Quang Vũ](#)⁷⁶, năm đấy rất 'đầu gấu' thì cũng chỉ suýt soát được huy chương thôi. Đối với Việt Nam hay rộng ra đối với Đông Nam Á thì như vậy là cực kỳ tốt rồi. Ở Châu Á lúc đấy có những đối thủ rất mạnh, nếu trừ Trung Quốc ra thì còn có Nhật, Hàn, Đài Loan. Đài Loan thì thường xuyên góp mặt World Finals chỉ có một trường duy nhất là Đại học Quốc gia Đài Loan thôi, còn Hàn Quốc thì có 1-2 trường, Nhật thì đông đội hơn.



Đội tuyển Chicken - Đại học Công nghệ ĐHQG Hà Nội có mặt trong Chung kết ICPC WF tại Tokyo - Nhật Bản gồm Trần Thị Thùy Trang, Phan Đa Phúc và Lê Huy Bình, Coach Bùi Thế Duy.

Năm Việt Nam mình lần đầu tiên được huy chương ICPC ở Dhaka, Bangladesh. Lúc đấy thì bác có cảm xúc thế nào?



Đội tuyển EggCentroy - Đại học Công nghệ ĐHQG Hà Nội có mặt trong ICPC WF tại Dhaka - Bangladesh gồm Bùi Hồng Đức, Nguyễn Hải Bình, Vũ Hoàng Kiên, Coach Hồ Đắc Phương

cho Singapore. Với việc Việt Nam khởi đầu với một tấm huy chương Đồng thì mong rằng trong tương lai gần Việt Nam sẽ đạt được cái huy chương Bạc tiếp theo.

Ta cũng phải nhìn nhận thực tế và nói rõ luôn, có những trường luôn đứng ở đỉnh cao liên tục rất

Thật ra mà nói thì việc nước mình tiệm cận với huy chương ICPC là đã có từ năm của Sỹ Nguyên rồi, lúc đấy mình đã nhận thức được là gần chạm tới rồi. Đến năm của Bùi Hồng Đức thì điều ấy càng trở nên chắc chắn hơn, một điều cũng may mắn là năm ấy vì tình hình chính trị, chiến tranh nên cũng mất khá nhiều đội mạnh làm cho việc có huy chương càng chắc chắn hơn. Ở năm ấy đi thi thì mọi người trong đoàn hy vọng đội sẽ được huy chương Bạc cơ, vì có khoảng thời gian đúng rất sát khoảng được Bạc, nhưng mà càng về phút cuối thì cứ rơi hạng dần và nằm trong ngưỡng cuối của huy chương Đồng.

Sau việc đấy thì mọi người đều rất vui và tiếp thêm cho Việt Nam nhiều hy vọng để chiến đấu hơn. nhưng mà sau này cũng chỉ một mình bạn Bùi Hồng Đức năm vừa rồi lại mang thêm một huy chương nữa về

⁷⁵<https://cphof.org/profile/codeforces:user202729>

⁷⁶<https://cphof.org/profile/topcoder:ConanKudo>



nhiều năm mà chưa bao giờ vô địch như Đại học Tokyo của Nhật Bản, luôn đứng ở mức Vàng hoặc sát Vàng, có những năm sát vô địch nhưng đều chưa được, kể cả Đài Loan cũng thế. Đây là một chặng đường rất là khó. Giá trị ICPC mang lại nhiều nhất ngoài vị thế của đất nước đây còn là tuyển chọn những bạn giỏi, ICPC cũng luyện cho sinh viên Việt Nam một điểm yếu cố hữu đây là làm việc nhóm - tập thể và cái tính hoàn thiện. Theo thời gian thì các đội từ một bài rồi giải được số bài tiệm cận đến mức huy chương cho ta thấy tính làm việc tập thể trong một đội dần tốt lên. Và đương nhiên ta cũng thấy một điều Việt Nam tốt lên hẳn đây chính là ngoại ngữ. Ở thời gian đầu Regional Việt Nam thì ban tổ chức còn xin hội đồng ICPC Châu Á cho phép làm ra hai bộ đề, một bộ đề Tiếng Việt cho các đội Việt Nam và bộ đề Tiếng Anh cho các đội nước ngoài, các nước như Hong Kong các thứ họ lại kiện mình 'chơi ăn gian'. Lập tức sau đó là năm 2009 mình chỉ làm đề Tiếng Anh. Các năm đầu còn thấy các đội Việt Nam mang từ điển chất đầy phòng thi, nhưng mấy năm nay có thầy mang từ điển nữa đâu! Còn cái thứ hai mà ICPC giúp sinh viên Việt Nam phát triển đây là tính hoàn thiện, do ICPC chỉ là 'một sai, hai đúng', nên các đội phải thật hoàn thiện bài làm và kiểm thử rất nhiều trường hợp trước khi nộp, nó khác hẳn với IOI là làm tính điểm, đúng đến đâu có điểm đến đó chứ không cần đúng hết.

Nhìn vào kết quả của những năm gần đây, theo bác những năm tới thì Tin học Việt Nam cần phát triển như nào để có thể lên cao hơn?

Phong trào Tin học cần phải được phổ biến hơn. Tất nhiên là thi đấu ở những cuộc thi đỉnh cao như thế này thì cần nhiều yếu tố rất phức tạp, nhưng hiện nay Việt Nam cũng còn nhiều định hướng để phấn đấu và cải thiện, trước tiên ta phải nhìn từ phái IOI luôn phải duy trì thành tích ổn định qua các năm để đảm bảo các thế hệ sau sẽ tiếp bước các thế hệ trước, ICPC thì rõ ràng ta phải tìm cách vượt qua những đối thủ hàng đầu Châu Á như Đài Loan, tiếp theo là Hàn, Nhật Bản, chưa kể là Trung Quốc, đây là những nước đang được đánh giá là số 1 toàn cầu. Còn các đội Mỹ vốn toàn là người gốc Trung Quốc nên mình cũng không nói làm gì (cười). Đây là các nước chúng ta phấn đấu để vươn tới, thời gian này cũng là cơ hội tốt để vượt qua khối Nga và Đông Âu, xét trong Đông Nam Á hay khu vực Ấn Độ thì mình vẫn ở vị trí ok, chưa có vấn đề gì cả!

Ta cũng thấy là chính từ ICPC cũng mở ra rất nhiều cơ hội cho Việt Nam, đây là sau khi Việt Nam hội nhập thì những cơ hội thực tập, làm việc cho các công ty lớn ở nước ngoài hay unicorn (kỳ lân công nghệ) là nhiều hơn hẳn. Điều này cũng cho thấy một cái khập khiễng ở Việt Nam đây là doanh nghiệp Việt Nam không mấy mặn mà với các bạn giỏi, bảo không cần thì cũng không dám nói, nhưng mà họ có mặn mà với những người đây không thì chưa chắc. Họ có thể nhiệt tình tìm kiếm, nhưng mà phía lãnh đạo chưa chắc đã nhận ra để tận dụng tài năng của họ, thật sự là như thế! Chính vì vậy những bạn giỏi nhất thì thường rẽ nhánh sang các công ty nước ngoài để làm, hoặc là các unicorn, có một số bạn thì tự lập start-up, như Lợi Lưu (CEO Kyber Network) hay Phạm Hữu Ngôn (CEO AhaMove) là những trường hợp như vậy. Thật ra nếu Việt Nam có một nền tảng doanh nghiệp lớn kiểu như Trung Quốc có Huawei để tận dụng các bạn giỏi, thì không biết thế nào nữa, có thể CNTT nước mình sẽ phát triển nhanh hơn. Hầu hết những bạn giỏi ở Việt Nam thì hết năm học thứ nhất là đã chuyển sang các trường như NTU, NUS hay các trường nước ngoài khác. Những bạn đây thì việc tiếp cận doanh nghiệp quốc tế sẽ mạnh hơn các bạn trong nước, ở những công ty mà cần những người có kỹ năng thật sự. Việc nhiều bạn giỏi ra nước ngoài làm cũng là mở đường kết nối cho các bạn trong nước.

Cái thứ nhất thì trên chặng đường tiếp theo thì mục tiêu của ta còn nhiều, ta luôn phải phấn đấu để



Đoàn Việt Nam tại ICPC WF Luxor - Ai Cập năm 2023



đạt được mục tiêu. Cái thứ hai là phong trào Tin học từ năm 2015-2017, nước mình là một trong những nước đầu tiên mở rộng Olympic Sinh viên đến với học sinh cấp 3. Và năm vừa rồi là có thử nghiệm cho cả dưới cấp 3 đi thi luôn, là bạn Nguyễn Khánh Phúc, mới lớp 9 thôi mà đi thi đạt kết quả cũng tốt, nếu Việt Nam cho phép học sinh cấp 2 đi thi HSGQG thì biết đâu bạn ý lại thi IOI được, đây cũng là trường hợp mở đường. Từ phong trào này thì mình bắt đầu chuyển sang Tin học trẻ, cũng do bên Hội Tin học đứng ra tổ chức. Tin học trẻ ngày xưa thì Trung ương đoàn cử người làm hết, nhưng giờ đây thì cũng có nhờ bên mình nhiều nên chất lượng cũng cải thiện dần. Với phong trào phát triển Tin học dạo gần đây thì các bạn trong VNOI đều biết rồi, từ hệ thống thi do các bạn VNOI tự làm, tự phát triển và tự xây dựng theo quan niệm lúa trước dạy lúa sau, thì công tác làm đề cũng ngày càng sát với đề quốc tế hơn, chất lượng kỳ thi ngày càng nâng cao hơn. Chứ nếu cứ dựa vào những người ra đề cũ, hệ thống thi cũ thì đã chẳng có những bộ đề chất lượng như vậy, đặc biệt trong Olympic thì luôn đề cao tính minh bạch, trung thực, mọi cái đều làm việc trực tiếp nên chúng ta làm việc dễ hơn, cởi mở hơn và không ai lẩn tẩn đến những khó khăn có thể ảnh hưởng đến phong trào chung. Mọi người đều mong muốn một sân chơi tốt cho các bạn trẻ Việt Nam được cọ xát và cạnh tranh, điển hình như VNOI CUP ấy, chắc gì những người ra đề đã giỏi hơn người thi, nhưng tất cả mọi người đều vui vẻ và tôn trọng nhau, bởi vì chúng ta đảm bảo được cái tính nghiêm túc và thử thách để mọi người cùng thi đấu. Từ những cơ sở đấy thì lúa sau cứ kế thừa và tiếp tục phát triển, một bài toán tính tiền rất rõ ràng!

Đúng là Tin học Việt Nam đến hôm nay đang có sự chuyển giao về mặt kiến thức, khi khóa trước sẽ dạy cho khóa sau và chất lượng đề thi cũng ngày càng tốt lên

Cũng nói thật với các bạn luôn, trước đây đề thi HSGQG cũng là những thầy cũ ra với nhau, về sau thì hầu hết ban ra đề đa dạng và là người xuất phát từ thi Olympic hay từ Hội Tin học ra đề hết. Hiện nay, số lượng các bạn trẻ trong ban ra đề cũng rất là nhiều. Các kỳ thi của Bộ Giáo dục và Đào tạo rồi kể cả là Tin học trẻ cũng dựa hết vào ban ra đề này, việc tiến cử các bạn trẻ thì do ban tổ chức Olympic và ICPC làm. Ta cũng thấy Tin học trẻ bây giờ mức độ đề đã khó hơn rất nhiều, cũng chính vì chất lượng đề tốt lên nên Tin học trẻ phải bỏ chữ "không chuyên", ngày xưa là có Tin học trẻ không chuyên nữa. Trước đây thì Hội tin học đã tập trung nhiều vào nhóm cấp 3 và Đại học rồi nên bây giờ sẽ tập trung vào nhóm cấp 2 để đẩy trình độ chung của nhóm đấy lên. Toàn bộ hệ thống thi Học sinh giỏi và nguồn thông tin tài năng về lập trình thì hiện nay là do chúng ta đang nắm giữ.

Bác đã có nhiều năm tiếp xúc với rất nhiều thế hệ sinh viên Tin học tài năng, rất nhiều tấm gương từ thành công đến vô cùng thành công, bác có lời khuyên nào cho các bạn trẻ về việc có thể tiến xa như các bậc đàn anh đi trước được không?

Nếu nói riêng về cộng đồng VNOI thôi thì hiện các bạn cũng như các anh như đi trước như Lợi Lưu, Phạm Hữu Ngôn, vậy, nhưng các bạn là những lúa trẻ hơn và kĩ năng hơn. Dù là ai thì cũng đều mong muốn có một sản phẩm để đời hay một đóng góp có ích cho xã hội, trước tiên đấy là việc 'thành người'. Còn về 'lập nghiệp' thì cũng chia ra 2 hướng thôi, một là nghiên cứu - giảng dạy, dạng theo hướng nghiên cứu giảng dạy thì đã có rất nhiều tấm gương thành công như thầy Bùi Thế Duy, Phạm Bảo Sơn, Trần Minh Triết, ... hoặc đi dạy Tin học thì có Phạm Văn Hạnh. Có những người thích như vậy, còn cái thứ hai là bằng tài năng của mình sẽ theo hướng công nghiệp là đi làm. Trong công nghiệp thì có rất nhiều lựa chọn, thường thì sẽ phải tôi luyện ở nhiều môi trường khác nhau như start-up hay là big tech (các tập đoàn lớn) để có cơ hội xử lý những bài toán lớn cần dùng nhiều đến kỹ năng lập trình và thuật toán, giải quyết những bài toán tối ưu. Khi tích lũy đủ kinh nghiệm thì sau đó có thể lập nghiệp và khởi nghiệp. Đây là con đường tôi thấy rất nhiều, và cũng chứng minh luôn, người mà thành công nhất tôi từng thấy là Nguyễn Thành Trung (CEO Sky Mavis).

Sau khi thi World Finals xong thì Trung có bắt đầu một cái mạng xã hội để chia sẻ đồ ăn là Lozi. Sau đó là do sản phẩm không muốn đi theo hướng phát triển công nghệ mà chỉ muốn kiếm tiền nên anh ấy bắt đầu chán, bỏ công ty đấy và nghỉ một thời gian để nghĩ ra cái mới. Đây là điểm bắt đầu của Sky Mavis. Hoặc là như nhóm của Lợi Lưu với Kyber Network hay như Phạm Hữu Ngôn khi ra trường cũng kiên quyết theo hướng khởi nghiệp luôn, đốt một số triệu đô lập nghiệp và bây giờ cũng rất thành công với



AhaMove. Bây giờ Ngôn chắc code kém rồi nhưng gọi được vốn trăm triệu đô của Temasek cũng không phải là dạng vừa (cười). Hay như [Lê Yên Thanh](#)⁷⁷ thì vẫn say mê với Busmap. Nhưng mà chung quy lại thì các bạn trẻ vẫn phải tìm những cái doanh nghiệp mà có định hướng như có những bài toán công nghệ, bài toán đủ khó để mà mình dần thân thì tốt hơn, cái điều này cũng quay lại câu chuyện lúc nãy, ở Việt Nam thì lại không sẵn sàng cho những thứ đó, nên nếu các bạn muốn đi đúng lại phải tìm những cái luồng từ nước ngoài về. Có một số bạn cũng sẽ phản biện là Việt Nam cũng có FPT rất to, Viettel cũng rất lớn, còn VNPT rất oách, thật sự tôi cũng đã kiểm tra các bạn thi Olympic ngày xưa có thứ hạng cao mà làm ở đây thì chưa thấy ai thực sự thành công. Thực ra là không phải họ không có khả năng thành công, bởi vì họ không tiếp cận được những cái bài toán đủ tầm, để mình có thể cống hiến được bằng cái năng lực mà mình đã có được qua việc trải qua các cái kì thi này. Đây là cái mà tôi không cảm nhận được. Tại vì, cái thành công ở đây tôi muốn là cái thành công về mặt công nghệ và năng lực công nghệ chứ không phải thành công về việc kiếm tiền. Chẳng ví dụ như là Shark Bình trong Shark Tank là một ví dụ đây, về thành công về mặt tiền bạc thì có thể, những thành công về mặt công nghệ thì chưa chắc.

Lúc này ta sẽ đặt ra câu hỏi cuối cùng, ở Việt Nam có cơ hội không?

Đầu tiên đi đồng hành cùng các bạn Olympic rất nhiều năm, ít nhất là cũng phải 20 năm và bản thân tôi cũng là người trong nhóm đấy nếu xét từ thời chưa có Tin học. Ước mơ của tất cả mọi người như chúng ta là có thể làm được một điều lớn lao từ năng lực chuyên môn của bản thân. Vậy nhưng môi trường có cho phép chúng ta làm được điều đấy không? Điều này rất đáng cân nhắc, cũng rất may là sau khi Việt Nam hội nhập với thế giới thì tất cả các bạn đều có cơ hội, làn sóng các bạn làm ở big tech như Facebook, Amazon, Google đều tràn về Việt Nam rất nhiều, ngược lại thì làn sóng rời Việt Nam để làm cho các công ty xuyên biên giới cũng không ít, có những bạn thậm chí chả cần rời Việt Nam nhưng vẫn ngồi ở nhà làm việc cho doanh nghiệp nước ngoài tốt, được làm việc với công nghệ ở nước ngoài thì đây là cơ hội không thể tốt hơn. Còn công nghệ ở Việt Nam thì vừa rồi ta cũng phải cảm ơn Tổng Bí Thư khi đã nhìn nhận và nói rằng nhìn số liệu và thứ hạng của Việt Nam trong lĩnh vực CNTT thì rất cao và hoành tráng, nhưng đáng tiếc là chúng ta chỉ đang làm phần thừa hay phần thấp nhất trong chuỗi cung ứng chứ chưa được làm những phần quan trọng nhất và hi vọng rằng với những chính sách mới thì chúng ta sẽ có không phải làm những phần đấy nữa, các bạn trẻ sẽ có cơ hội đóng góp giá trị hơn về mặt công nghệ. Từ đó ta nhận ra một thực trạng là các doanh nghiệp Việt Nam cần trình độ như các bạn trong cộng đồng VNOI thì chưa nhiều lắm. Đây là cái chuyện mà nói đây nó cũng hơi động chạm một tí nhưng mà nó là vấn đề khó nhưng mà chúng tôi hy vọng nó sẽ đổi.

Vậy bác nhìn nhận về tình hình CNTT trên thế giới đang như thế nào?



Đoàn Việt Nam tại IOAI năm 2024

Ta có thể thấy cái làn sóng công nghệ của thế giới đang chia ra hai làn sóng là AI và Security sắp tới cũng sẽ có hai cái Olympic Quốc tế về hai cái đó. Hiện nay, Bộ Giáo dục và Đào tạo đều đang đề nghị là phía Hội Tin học đứng ra hỗ trợ. Thế nhưng Bộ Giáo dục cũng đang băn khoăn là đều thi Tin và lo ngại là sẽ giống như kỳ thi sáng tạo Khoa học kỹ thuật ấy, kiểu như thi ba người thì một người làm thôi còn hai người còn lại ăn theo nên họ cũng vẫn đang cân nhắc về khả năng tham gia của Việt Nam. Còn về phía trong nước thì năm tới chắc chắn sẽ mở ra thêm kỳ thi Olympic AI,

bởi vì đây là xu thế để các bạn có nhiều lựa chọn hơn, và AI đang là hướng yêu cầu kỹ năng của các bạn học sinh rất nhiều, theo dạng kỹ năng để làm AI chứ không phải các bạn chỉ cần vẽ ra một dự án AI rồi bảo nó hay hoặc không, kiếm nhiều tiền hay ít tiền, hoàn toàn làm việc bằng năng lực. Người thi phải biết rõ về dữ liệu, các thuật toán về dữ liệu, thuật toán về machine learning (học máy), deep learning (học sâu), ngay cả computer vision (thị giác máy tính) hay các mô hình ngôn ngữ lớn, đều là đỉnh thuật toán hết. Còn về blockchain thì nó còn nhiều thứ nữa, hiện nay thì môi trường giáo dục thế hệ trẻ của mình vẫn chưa chấp nhận, trong các văn bản vẫn chỉ tràn trề là công nghệ chuỗi khối - blockchain hoặc

⁷⁷<https://cphof.org/profile/topcoder:binhminh410>



là định nghĩa pháp lý về tài sản số, chứ chưa có những tri thức cụ thể. Chỉ khi nào đã có quy chế hay định nghĩa rõ ràng về công nghệ bởi giới công nghệ thì lúc đó Hội Tin học mới vào cuộc.

Nói đi cũng phải nói lại thì thứ rất là rõ ràng trong trào lưu khoảng 5-7 năm gần đây thì các bạn xuất sắc chuyển sang blockchain - smart contract và AI rất là nhiều, đặc biệt có những bạn chuyển sang quant trading là những nơi đòi hỏi về toán và thuật toán rất khủng khiếp.

Các bạn giỏi bây giờ lương càng ngày càng cao lên, tiếp cận với công nghệ mới sớm hơn.

Đúng vậy, giỏi luôn phải đi đôi với có đầu óc!

Rất là cảm ơn bác đã tham gia buổi phỏng vấn hôm nay. Thay mặt VNOI và Cộng đồng Tin học Việt Nam chúng cháu rất là cảm ơn bác đã dành rất nhiều công sức cho Tin học Việt Nam trong rất nhiều năm. Mong bác tiếp tục có nhiều sức khỏe, nhiều minh mẫn để tiếp tục đồng hành với Cộng đồng.

Lời cuối cùng, với phong trào Olympic, ICPC hay kể cả về sau là Olympic AI ở Việt Nam thì tôi cũng phải đồng hành cùng các bạn. Theo quan điểm là những người đi trước thì tích lũy và để hình thành ra một cái lứa đàn em giỏi hơn mình, mạnh hơn mình và lại đẩy tiếp cho cả cộng đồng tiến lên. Cái đấy là cái mà chúng tôi phải cảm ơn ngược lại thế hệ trẻ. Tôi cũng cảm thấy là hai bên đều có sự đồng cảm, sự thống nhất về lộ trình: chúng ta sẽ đi tìm được cái đích mà nó tốt hơn, đẹp hơn, có ý nghĩa hơn đối với cuộc sống của đất nước Việt Nam.

Vâng, cảm ơn bác rất là nhiều!

LỜI CẢM ƠN

BIÊN TẬP

Nguyễn Trung Quân (Tổng biên tập)
Nguyễn Tấn Minh (Phó Tổng biên tập)
Bùi Nguyễn Lan Vy
Nguyễn Hoàng Gia Hân
Phan Vĩnh Tiến
Nguyễn Đăng Quân

THIỆT KẾ

Đoàn Viết Tiến Đạt
Trần Thị Thanh Vân

CỔ VẤN

Phạm Thị Diễm Quỳnh
Vương Hoàng Long
Trần Quang Lộc
Phạm Xuân Trung
Nguyễn Minh Nhật
Nguyễn Đức Huy
Lê Đức Anh

CỘNG TÁC VIÊN

Trần Xuân Bách (tác giả bài viết)
Võ Đức Đoàn (tác giả bài viết)



TẠP CHÍ VNOI XUÂN ẤT TỴ 2025
XUẤT BẢN NĂM 2025