



Thời gian làm bài: 240 phút

Ngày thi: 10-12-2025

TỔNG QUAN ĐỀ THI

Bài 1. Số đẹp – SPRIME (100 điểm)	2
Bài 2. Tối ưu độ trễ – LATENCYOPT (100 điểm)	3
Bài 3. Tìm ếch – XFROG (100 điểm)	5
Bài 4. Chia bảng - LEAGUE (100 điểm)	7

Lưu ý:

- Điểm của các bài 1,2,3 là tổng điểm của các subtask. Điểm của subtask với mỗi lần nộp là điểm nhỏ nhất đạt được của các test trong subtask đó;
- Thí sinh không được phép sử dụng các định hướng biên dịch chương trình có các từ khoá sau: `pragma`, `optimize`, `target`, `O3`, `Ofast`, `unroll-loops`, `avx`, `avx2`, `fma`, `omit-frame-pointer`.

Bài 1. Số đẹp – SPRIME

Hạn chế thời gian: 1 second
Hạn chế bộ nhớ: 1024 megabytes

Ta gọi một số nguyên dương là *đẹp* nếu tổng các chữ số (ở dạng biểu diễn thập phân) của nó là một số nguyên tố. Ví dụ 1112 là một số đẹp, còn 2025 thì không.

Viết tất cả các số đẹp không quá 10^{15} liên tiếp nhau theo thứ tự tăng dần, tạo thành một chuỗi S . Các ký tự của S được đánh số bắt đầu từ 1. Đoạn đầu của chuỗi S là: 2357111214162021...

Cho một chuỗi x có độ dài không quá 100, chỉ chứa các chữ số. Hãy tìm vị trí nhỏ nhất mà x xuất hiện như một chuỗi con liên tiếp trong S .

Dữ liệu

- Dòng đầu chứa số nguyên T ($1 \leq T \leq 100$) — số lượng testcase.
- Mỗi testcase gồm một dòng chứa chuỗi x . Dữ liệu bảo đảm x là chuỗi con liên tiếp của S .

Kết quả

Với mỗi test, in ra vị trí nhỏ nhất mà x xuất hiện trong S .

Ví dụ

stdin	stdout
3	5
11	1
235	8
21	

Cách tính điểm

Gọi n là độ dài chuỗi x .

- Subtask 1 (25 điểm): $n \leq 3$.
- Subtask 2 (25 điểm): $3 < n < 30$.
- Subtask 3 (50 điểm): $30 \leq n \leq 100$.

Bài 2. Tối ưu độ trễ – LATENCYOPT

Hạn chế thời gian: 1 seconds
Hạn chế bộ nhớ: 1024 megabytes

Một nhà cung cấp dịch vụ Internet vận hành hệ thống gồm N trạm, được nối với nhau bằng $N - 1$ tuyến cáp quang tạo thành một mạng có dạng cây. Mỗi tuyến cáp nối hai trạm U, V có giá trị W biểu thị độ trễ khi một gói tin đi qua tuyến đó. Độ trễ truyền tin giữa hai trạm bất kỳ là tổng độ trễ của các tuyến cáp trên đường truyền tin giữa hai trạm đó. **Độ trễ của mạng** là độ trễ truyền tin lớn nhất trong tất cả các cặp trạm.

Để cải thiện chất lượng dịch vụ, nhà mạng đưa ra Q phương án, mỗi phương án là thay thế một dãy các trạm liên tiếp bằng một trạm mới. Cụ thể, trong phương án thứ i ($1 \leq i \leq Q$), nhà mạng chọn hai trạm A_i và B_i , xác định đường truyền tin duy nhất giữa chúng, và dự định xây một trạm mới để thay thế toàn bộ các trạm trên đường truyền tin này. Mỗi tuyến cáp từ các trạm bên ngoài nối đến các trạm bị thay thế sẽ được chuyển sang trạm mới, với độ trễ tương ứng như ban đầu.

Để đánh giá mức độ hiệu quả của từng phương án trước khi triển khai, nhà mạng cần xác định độ trễ của mạng nếu áp dụng phương án đó. Lưu ý, các phương án chỉ đang trong quá trình đánh giá độ hiệu quả, chưa có phương án nào thực sự được thực thi.

Dữ liệu

Dòng đầu tiên chứa số nguyên T là số lượng test. Mỗi test được mô tả như sau:

- Dòng đầu chứa số nguyên N ($1 \leq N \leq 2 \times 10^5$).
- $N - 1$ dòng tiếp theo, mỗi dòng gồm ba số nguyên U, V, W ($1 \leq U, V \leq N; 1 \leq W \leq 10^9$) mô tả một tuyến cáp quang nối hai trạm U và V với độ trễ W .
- Dòng tiếp theo chứa số nguyên Q ($1 \leq Q \leq 2 \times 10^5$).
- Dòng thứ i trong số Q dòng tiếp theo chứa hai số nguyên A_i, B_i ($1 \leq A_i, B_i \leq N; A_i \neq B_i$) mô tả một phương án.

Gọi S_N, S_Q lần lượt là tổng N và tổng Q trong tất cả các test, dữ liệu bảo đảm $S_N, S_Q \leq 2 \times 10^5$.

Kết quả

- Với mỗi phương án, in ra một số nguyên trên một dòng là độ trễ của mạng nếu áp dụng phương án đó.

Ví dụ

standard input	standard output
1	12
7	13
1 2 4	11
2 3 7	
2 4 2	
4 5 6	
4 6 3	
6 7 5	
3	
5 3	
1 7	
3 6	

Cách tính điểm

- Subtask 1 (6 điểm): $S_N, S_Q \leq 5000$;
- Subtask 2 (25 điểm): Với mọi phương án, A_i và B_i được nối trực tiếp bởi một tuyến cáp;
- Subtask 3 (25 điểm): Với mọi phương án, đường đi giữa A_i và B_i chứa không quá 20 trạm;
- Subtask 4 (23 điểm): $A_i = 1$;
- Subtask 5 (21 điểm): Không có ràng buộc gì thêm.

Bài 3. Tìm ếch – XFROG

Hạn chế thời gian: 2 giây
Hạn chế bộ nhớ: 1024 megabytes

Có một chú ếch đang trốn trong một rừng cây, có thể mô tả bằng một trục số. Vị trí ban đầu của ếch ở tọa độ nguyên dương s . Chú ếch có sức nhảy cố định k là một số nguyên dương. Cả hai giá trị s và k bạn đều không biết, chỉ biết rằng s và k đều không quá 10^9 .

Bạn được trang bị một *máy dò nhiệt*. Mỗi lần sử dụng, bạn chọn một đoạn $[L, R]$ trên trục số (với L, R là các số nguyên dương, $1 \leq L \leq R \leq 10^{18}$). Máy dò sẽ cho bạn biết **tại thời điểm đó** chú ếch có đang ở trong đoạn $[L, R]$ hay không.

Tuy nhiên, mỗi lần dò như vậy lại gây ra tiếng ồn làm ếch hoảng sợ và sẽ nhảy sang phải một đoạn đúng bằng k đơn vị. Nghĩa là, nếu trước khi dò, vị trí của ếch là p , thì sau khi nhận kết quả từ máy dò, vị trí mới của ếch sẽ là $p + k$.

Yêu cầu: Thiết kế chiến lược tương tác để xác định đúng vị trí hiện tại của chú ếch trong không quá q lần dò.

Tương tác

- Ban đầu bạn cần đọc vào một số nguyên τ ($1 \leq \tau \leq 7$) là số thứ tự subtask của test hiện tại (xem thêm ở phần chấm điểm).
- Mỗi lần hỏi, bạn cần in ra một dòng có dạng: `? L R` ($1 \leq L \leq R \leq 10^{18}$) sau đó đọc từ đầu vào chuẩn một xâu YES/NO tương ứng là tại thời điểm dò, ếch có đang ở trong phạm vi $[L, R]$ hay không.
- Khi đã chắc chắn tìm được vị trí hiện tại của ếch, in ra một dòng: `! f` trong đó f là số nguyên trả lời cho **vị trí hiện tại** mà ếch đang đứng. Sau khi trả lời xong, bạn cần kết thúc chương trình.

Lưu ý:

- Cần flush bộ đệm sau mỗi lần in ra để tương tác được với trình chấm (ví dụ dùng `fflush(stdout)` hoặc `cout << endl`; trong C++).
- Nếu bạn hỏi quá q lần, trình chấm sẽ dừng ngay lập tức, bạn cần dừng chương trình và nhận kết quả **Wrong Answer**.

Cách tính điểm

Trong tất cả các test, $1 \leq s \leq 10^9$, $1 \leq k \leq 10^9$.

- Subtask 1 (19 điểm): $k = 1$, $q = 80$
- Subtask 2 (11 điểm): $s \leq 1000$, $k \leq 1000$, $q = 80$
- Subtask 3 (12 điểm): $s \leq 1000$, $k \leq 1000$, $q = 22$
- Subtask 4 (13 điểm): $s \leq 10^6$, $k \leq 10^6$, $q = 80$
- Subtask 5 (14 điểm): $s \leq 10^6$, $k \leq 10^6$, $q = 42$
- Subtask 6 (15 điểm): $s \leq 10^9$, $k \leq 10^9$, $q = 80$
- Subtask 7 (16 điểm): $s \leq 10^9$, $k \leq 10^9$, $q = 62$

Ví dụ

stdin	stdout	Giải thích
2		Số thứ tự subtask của test hiện tại
	? 1 4	Dò xem tại thời điểm này ếch có ở trong đoạn $[1, 4]$ không
NO		Giả sử lúc này ếch đang ở vị trí 5, nên trả lời NO
	? 5 8	Dò tiếp đoạn $[5, 8]$
YES		Giả sử ở thời điểm dò này ếch đang ở vị trí 8, nên trả lời YES
	? 9 12	Dò đoạn $[9, 12]$
YES		Giả sử ở thời điểm dò này ếch đang ở vị trí 11, nên trả lời YES
	! 14	Trả lời vị trí hiện tại của ếch

Ví dụ trên chỉ minh họa cách chương trình của bạn và trình chấm tương tác với nhau, không thể hiện quá trình suy luận đáp án.

Trong ví dụ này, giả sử ếch có vị trí ban đầu $s = 5$ và độ dài bước nhảy $k = 3$.

- Trước lần dò 1, ếch ở 5. Dò $[1, 4]$ nhận kết quả NO. Sau đó ếch nhảy đến 8.
- Trước lần dò 2, ếch ở 8. Dò $[5, 8]$ nhận kết quả YES. Sau đó ếch nhảy đến 11.
- Trước lần dò 3, ếch ở 11. Dò $[9, 12]$ nhận kết quả YES. Sau đó ếch nhảy đến 14.

Vì vậy khi chương trình đã xác định được s và k , và biết mình đã thực hiện 3 lần dò, thì vị trí hiện tại của ếch là $f = s + 3k = 14$, nên in ra ! 14.

Bài 4. Chia bảng - LEAGUE

Ban tổ chức đang chuẩn bị chia bảng cho một giải bóng đá có n đội, đánh số từ 1 đến n . Mỗi đội bóng được mô tả bằng t chỉ số *kỹ năng* (ví dụ: tấn công, phòng ngự, thể lực, ...). Các chỉ số kỹ năng của các đội đã được chuẩn hoá về cùng đơn vị là các số nguyên dương không vượt quá 10^9 .

Giữa mỗi cặp đội bóng i, j cũng đã biết trước *khoảng cách di chuyển* $d_{i,j}$ nếu hai đội nằm cùng bảng và phải đá sân nhà – sân khách. Ta có $d_{i,j} = d_{j,i}$ và $d_{i,i} = 0$.

Mục tiêu của ban tổ chức là chia các đội bóng thành k bảng, sao cho:

- Mỗi đội thuộc một bảng duy nhất.
- Mỗi bảng có ít nhất 2 đội.
- Trong mỗi bảng, với kỹ năng thứ s ($1 \leq s \leq t$), tổng chỉ số kỹ năng đó của các đội trong bảng đó đều phải nằm trong giới hạn $[L_s, R_s]$.

Cần tìm một phương án chia bảng sao cho tổng khoảng cách di chuyển giữa tất cả các cặp đội cùng bảng là càng nhỏ càng tốt.

Đây là một bài toán tối ưu khó, vì vậy đề bài được cho dưới dạng **output only**: bạn được cung cấp sẵn các file *.inp chứa dữ liệu, và nhiệm vụ của bạn là tạo ra các file *.out mô tả một cách chia bảng sao cho thoả mãn các ràng buộc và có tổng khoảng cách di chuyển càng nhỏ càng tốt.

Dữ liệu

Mỗi file input gồm:

- Dòng đầu chứa ba số nguyên n, k, t :
 - $1 \leq n \leq 500$: số đội bóng.
 - $2 \leq k \leq 20$: số bảng cần chia.
 - $2 \leq t \leq 5$: số chỉ số kỹ năng trên mỗi đội.
- n dòng tiếp theo, dòng thứ i ($1 \leq i \leq n$) mô tả đội bóng thứ i :
 - Đầu tiên là t số nguyên $a_{i,1}, a_{i,2}, \dots, a_{i,t}$ — các chỉ số kỹ năng của đội i ;
 - tiếp theo là n số thực $d_{i,1}, d_{i,2}, \dots, d_{i,n}$ — khoảng cách di chuyển từ đội i tới từng đội $1, 2, \dots, n$ trong giải đấu. Ma trận khoảng cách thoả mãn $d_{i,j} = d_{j,i} \geq 0$ và $d_{i,i} = 0$.
- Dòng tiếp theo chứa t số nguyên $L_1, L_2, L_3, \dots, L_t$
- Dòng cuối cùng chứa t số nguyên $R_1, R_2, R_3, \dots, R_t$

Kết quả

File output tương ứng với mỗi file input của ban tổ chức có dạng:

- Gồm k dòng, dòng thứ j ($1 \leq j \leq k$) liệt kê các chỉ số đội bóng thuộc bảng thứ j , phân tách nhau bằng đúng một dấu cách. Mỗi đội từ 1 đến n phải xuất hiện **chính xác một lần** trong toàn bộ k dòng này.

Các bảng có thể được liệt kê theo bất kỳ thứ tự nào. Trong mỗi dòng, thứ tự các đội trong cùng một bảng là tùy ý.

Ví dụ

Dữ liệu	Kết quả
4 2 2 3 5 0 4 5 7 7 2 4 0 8 3 4 3 5 8 0 9 5 9 7 3 9 0 7 6 12 14	1 3 2 4

Trong ví dụ trên, các đội $\{1, 3\}$ được xếp vào bảng thứ nhất, các đội $\{2, 4\}$ vào bảng thứ hai. Tổng của từng loại kỹ năng thứ nhất trong mỗi bảng đều nằm trong giới hạn $[7, 12]$, kỹ năng thứ hai trong mỗi bảng đều nằm trong giới hạn $[6, 14]$ và giá trị hàm mục tiêu F (tổng khoảng cách di chuyển giữa các cặp đội cùng bảng) bằng 8.

Cách tính điểm

Với mỗi test, giả sử F_{jury} là giá trị nhỏ nhất trong đáp án của ban giám khảo, và F_{you} là giá trị F tương ứng với cách chia bảng trong file output của bạn.

- Nếu $F_{\text{you}} \leq F_{\text{jury}}$, bạn được 100% số điểm.
- $F_{\text{you}} > F_{\text{jury}}$, tỷ lệ điểm cho test đó được tính theo công thức

$$40\% + 60\% \times \left(\frac{F_{\text{jury}}}{F_{\text{you}}}\right)^{10}$$

Nếu file output không hợp lệ thì điểm cho test đó bằng 0.

- Điểm cuối cùng là tổng điểm các test.