



# OLYMPIC TIN HỌC SINH VIÊN VIỆT NAM LẦN THỨ 33

Từ ngày 10-13/12/2024, tại Trường Đại học Công nghiệp Hà Nội



## KHỐI SIÊU CUP

*Thời gian làm bài: 240 phút*

**Ngày thi: 11-12-2024**

### TỔNG QUAN ĐỀ THI

Bài 1. Biến đổi xâu — TRANSF (100 điểm) . . . . .	2
Bài 2. Ngôi nhà mới — NEWHOME (100 điểm) . . . . .	4
Bài 3. Trò chơi trên vòng tròn — CIRCLE (100 điểm) . . . . .	6
Bài 4. Chia xâu — BSTRING (100 điểm) . . . . .	7

*Lưu ý:*

- Điểm của mỗi bài 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. Biến đổi chuỗi — TRANSF

Alice có hai chuỗi  $S$  và  $T$ , mỗi chuỗi gồm các ký tự trong tập từ 'a' đến 'z'. Chuỗi  $S$  và chuỗi  $T$  có độ dài lần lượt là  $\ell_S$  và  $\ell_T$ . Alice muốn biến đổi chuỗi  $S$  và  $T$  để chúng giống nhau bằng việc chèn thêm các chuỗi khác vào các vị trí bất kỳ. Có  $K$  chuỗi cho trước (gọi là *chuỗi thêm*) được đánh số từ 1 đến  $K$ , mỗi chuỗi có thể được sử dụng không giới hạn số lần để Alice chèn vào chuỗi  $S$  và  $T$ .

Việc chèn chuỗi thêm vào một chuỗi  $Z$  (với  $Z$  là chuỗi  $S$  hoặc chuỗi  $T$ ) cần được thực hiện với các lưu ý sau:

- Chuỗi  $Z$  có độ dài  $\ell$  gồm các ký tự  $Z_1Z_2\dots Z_\ell$ . Mỗi vị trí chèn được đánh số từ 0 đến  $\ell$ . Vị trí 0 là vị trí ngay bên trái ký tự  $Z_1$ . Vị trí  $i$  ( $1 \leq i \leq \ell$ ) là vị trí ngay bên phải ký tự  $Z_i$ .
- Một chuỗi được chèn vào  $Z$  có thể thực hiện tại bất kỳ vị trí nào trong chuỗi, bao gồm trước ký tự đầu tiên và sau ký tự cuối cùng.
- Ta có thể chèn nhiều chuỗi vào cùng một vị trí. Các chuỗi thêm được chèn lần lượt theo một thứ tự xác định. Ví dụ, với  $Z = Z_1Z_2Z_3$ , ba chuỗi thêm  $A = A_1A_2A_3$ ,  $B = B_1B_2$  và  $C = C_1C_2C_3C_4$  được chèn lần lượt vào vị trí 1, ta thu được chuỗi  $Z_1C_1C_2C_3C_4B_1B_2A_1A_2A_3Z_2Z_3$ .
- Vị trí chèn các chuỗi là cố định kể cả sau khi đã chèn một số chuỗi. Ví dụ, với  $Z = \text{"aabc"}$ , nếu ta chèn "dd" vào vị trí 0; "e" và "gg" vào vị trí 1; "f" vào vị trí 4 thì ta có chuỗi kết quả là  $Z = \text{"ddaggeabcf"}$ .

**Yêu cầu:** Hãy giúp Alice chèn các chuỗi thêm sao cho chuỗi  $S$  giống với chuỗi  $T$  và tổng số chuỗi chèn thêm vào là ít nhất.

## Dữ liệu

- Dòng đầu tiên chứa chuỗi  $S$  ( $0 \leq \ell_S \leq 300$ ).
- Dòng thứ hai chứa chuỗi  $T$  ( $0 \leq \ell_T \leq 300$ ).
- Dòng thứ ba chứa số nguyên  $K$  ( $1 \leq K \leq 30$ ), là số lượng chuỗi có thể sử dụng để chèn.
- Mỗi dòng trong số  $K$  dòng tiếp theo chứa một chuỗi  $X_i$  ( $1 \leq i \leq K$ ), biểu diễn chuỗi thêm thứ  $i$ . Tổng độ dài của  $X_i$  không vượt quá 30.

## Kết quả

- Nếu không thể biến đổi để  $S$  giống  $T$ , ghi duy nhất một số  $-1$ . Ngược lại, ghi ra số lượng chuỗi cần phải chèn.
- Dòng thứ  $i$  trong số  $\ell_S + 1$  dòng tiếp theo ghi ra một số nguyên  $P_i$  là số lượng chuỗi được chèn ở vị trí  $i$  trong chuỗi  $S$ . Nếu  $P_i > 0$  thì  $P_i$  số tiếp theo ghi ra các thứ tự các chuỗi được chèn lần lượt ở vị trí  $i$ .
- Dòng thứ  $j$  trong số  $\ell_T + 1$  dòng tiếp theo ghi ra  $Q_j$  là số lượng chuỗi được chèn ở vị trí  $j$  trong chuỗi  $T$ . Nếu  $Q_j > 0$  thì  $Q_j$  số tiếp theo ghi ra các thứ tự các chuỗi được chèn lần lượt ở vị trí  $j$ .

## Ví dụ

stdin	stdout
aaa b 1 aab	-1
aad bb 3 bbd ad d	5 0 2 1 3 0 0 1 2 0 2 2 3

## Giải thích

Trong ví dụ thứ hai:

- Với xâu  $S$ , cần 2 phép chèn tại vị trí 1. Xâu "bbd" và "d" được chèn tại vị trí 1.
- Với xâu  $T$ , chèn "ad" tại vị trí 0 và chèn "ad" và "d" tại vị trí 2.
- Xâu kết quả của cả  $S$  và  $T$  đều là "adbbdad".

## Cách tính điểm

- Subtask 1 (30% số điểm): Độ dài tất cả các xâu  $X_i$  ( $1 \leq i \leq K$ ) bằng 1.
- Subtask 2 (30% số điểm): Các xâu  $S$ ,  $T$  và  $X_i$  ( $1 \leq i \leq K$ ) chỉ chứa ký tự 'a' và/hoặc 'b'. Độ dài tất cả các xâu  $X_i$  không quá 2.
- Subtask 3 (40% số điểm): Không có ràng buộc nào thêm.

## Bài 2. Ngôi nhà mới — NEWHOME

Ở ngôi làng nọ có  $N$  ngôi nhà, đánh số từ 1 đến  $N$ . Có  $N - 1$  con đường, mỗi con đường nối một cặp ngôi nhà và cho phép đi lại theo cả hai chiều. Hệ thống đường bảo đảm đi lại giữa mọi ngôi nhà.

Có  $N$  công dân sống trong làng, mỗi ngôi nhà có đúng một công dân. Công dân ở ngôi nhà thứ  $i$  thích màu  $C(i)$ , vì vậy người đó đã sơn màu  $C(i)$  cho ngôi nhà thứ  $i$ .

Trưởng làng muốn đổi chỗ ở cho người dân của mình. Ông chọn ra ngẫu nhiên một hoán vị  $P = P_1, P_2, \dots, P_N$  thoả mãn  $C(i) = C(P_i)$  với mọi  $i = 1, 2, 3, \dots, N$  (đây là yêu cầu về màu sắc). Các hoán vị thoả mãn đều có xác suất được chọn như nhau, và ông chỉ chọn trong số các hoán vị thoả mãn yêu cầu về màu sắc mà thôi. Tiếp đến, ông ra lệnh cho người ở ngôi nhà thứ  $i$  chuyển nhà tới ngôi nhà thứ  $P_i$ , khi đó người này vẫn được ở trong ngôi nhà có màu yêu thích của mình; độ dài quãng đường mà người này phải đi là số cạnh trên đường đi đơn giữa  $i$  và  $P_i$ . Chi phí để đổi nhà theo hoán vị  $P$  được tính bằng tổng độ dài đường đi của tất cả công dân. Hãy tính kỳ vọng chi phí đổi nhà.

**Nhắc lại:** Kỳ vọng của một biến ngẫu nhiên  $X$  rời rạc được tính bởi:  $\sum_{A \in \mathcal{P}} X(A) \times p(A)$  với  $\mathcal{P}$  là tập tất cả các biến cố có thể xảy ra,  $p(A)$  là xác suất xảy ra biến cố  $A$ , và  $X(A)$  là giá trị của biến ngẫu nhiên  $X$  khi xảy ra biến cố  $A$ . Cụ thể trong trường hợp của bài này:  $\mathcal{P}$  là tập tất cả các hoán vị thoả mãn yêu cầu về màu sắc;  $A$  là một hoán vị thuộc  $\mathcal{P}$ ;  $p(A)$  là xác suất mà trưởng làng chọn hoán vị  $A$ ; và  $X(A)$  là chi phí để đổi nhà theo hoán vị  $A$ .

### Dữ liệu

- Dòng đầu tiên chứa số nguyên dương  $N$ .
- Dòng thứ hai chứa  $N$  số nguyên không âm  $C(1), C(2), C(3), \dots, C(N)$ .
- Mỗi dòng trong số  $N - 1$  dòng tiếp theo chứa hai số nguyên dương  $u, v$  cho biết có một con đường nối giữa ngôi nhà thứ  $u$  và ngôi nhà thứ  $v$ .

### Kết quả

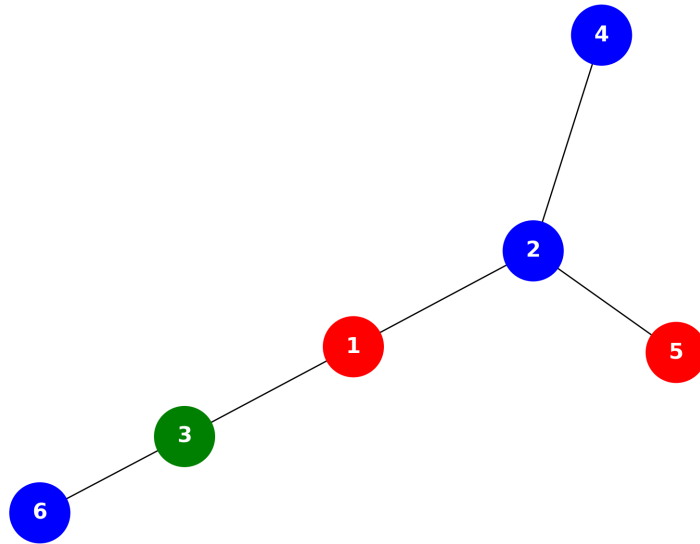
Kết quả là một số hữu tỷ, được biểu diễn dưới dạng phân số tối giản  $\frac{P}{Q}$ , hãy in ra  $P \times Q^{1000000005} \% 1000000007$ .

### Ví dụ

stdin	stdout
6	333333343
1 4 2 4 1 4	
1 2	
1 3	
2 4	
2 5	
3 6	

### Giải thích

Có 12 hoán vị thoả mãn, với tổng chi phí là 88.



## Cách tính điểm

- Trong tất cả các test:  $N, C(i) \leq 3 \times 10^5$ ;
- Subtask 1 (12 điểm):  $N \leq 10$ .
- Subtask 2 (16 điểm): Mỗi ngôi nhà chỉ có đúng một ngôi nhà khác có cùng màu với nó.
- Subtask 3 (20 điểm):  $N \leq 1000$ .
- Subtask 4 (24 điểm):  $C(1) = C(2) = \dots = C(n)$ .
- Subtask 5 (12 điểm): Không có ràng buộc nào thêm.

## Bài 3. Trò chơi trên vòng tròn — CIRCLE

Cho  $M$  tập số nguyên rời rạc, trong đó mỗi số nguyên từ 1 đến  $N$  thuộc duy nhất một trong  $M$  tập. Các tập  $1, 2, 3, \dots, M$  được sắp xếp liên tiếp trên một vòng tròn theo chiều kim đồng hồ (tập  $M$  kề với tập 1).

Thực hiện quy trình sau đây lặp đi lặp lại:

1. Tại mỗi bước, chọn số nhỏ nhất trong mỗi tập và loại bỏ số đó khỏi tập.
2. Số nhỏ nhất được loại bỏ khỏi tập  $i$  sẽ được thêm vào tập  $i + 1$  (với tập  $M$  thì thêm vào tập 1).
3. Nếu tập  $i$  rỗng thì không có số nào bị loại bỏ khỏi tập  $i$  và thêm vào tập kề  $i + 1$  (với số bị loại bỏ khỏi tập  $M$  thì thêm vào tập 1).

Có  $Q$  truy vấn, mỗi truy vấn yêu cầu xác định số nhỏ nhất bị loại bỏ tại lần chuyển thứ  $T$  của tập 1, nếu tại lần chuyển thứ  $T$  mà tập 1 không có số nào thì đưa ra  $-1$ .

### Dữ liệu

- Dòng đầu tiên chứa ba số nguyên dương  $N, M, Q$  ( $N, M, Q \leq 300\,000$ );
- Dòng thứ hai chứa  $N$  số nguyên dương  $G_1, G_2, \dots, G_n$  ( $1 \leq G_i \leq M$ ), với  $G_i$  là tập mà số  $i$  ban đầu thuộc;
- Dòng tiếp theo chứa  $Q$  số nguyên  $T_1, T_2, \dots, T_Q$  ( $1 \leq T_j \leq 10^9$ ) tương ứng với các truy vấn.

### Kết quả

Ghi ra kết quả trên một dòng gồm  $Q$  số tương ứng với kết quả của mỗi truy vấn.

### Cách tính điểm

- Subtask 1 (9 điểm):  $N, M, Q, T_j \leq 1\,000$ ;
- Subtask 2 (23 điểm):  $M \leq 1\,000$ ;
- Subtask 3 (35 điểm):  $Q = 1$ ;
- Subtask 4 (33 điểm): không có ràng buộc gì thêm.

### Ví dụ

stdin	stdout
5 3 2 1 1 2 2 3 1 2	1 2

## Bài 4. Chia xâu — BSTRING

Trong khoa học máy tính, biểu diễn và xử lý các số nhị phân là một phần quan trọng của nhiều bài toán thực tế. Một số nhị phân có càng nhiều bit 1 thì mức độ phức tạp khi lưu trữ hoặc xử lý càng lớn. Vì thế, việc tối thiểu hóa số lượng bit 1 trong các phép tính số nhị phân là một bài toán có ý nghĩa thiết thực.

Với một xâu nhị phân, bạn cần chia xâu này thành các đoạn liên tiếp. Mỗi đoạn sẽ được chuyển đổi thành một số nguyên dưới dạng nhị phân. Sau đó, tính tổng các đoạn này.

**Yêu cầu:** Tìm cách chia xâu sao cho tổng các đoạn đó, khi biểu diễn ở dạng nhị phân, chứa số lượng bit 1 ít nhất.

### Dữ liệu

- Dòng đầu tiên chứa số nguyên dương  $Q$  ( $Q \leq 100$ ) là số truy vấn.
- Mỗi dòng trong số  $Q$  dòng tiếp theo chứa xâu nhị phân trong truy vấn thứ  $i$ .

### Kết quả

Ghi ra  $Q$  dòng, dòng thứ  $i$  đưa ra tương ứng câu trả lời cho xâu ở truy vấn thứ  $i$ . Với mỗi xâu, thực hiện chèn dấu cộng vào các vị trí các điểm ngắt đoạn xâu. Không đưa ra dấu cách hoặc ký tự bất kỳ khác.

### Ví dụ

stdin	stdout
2	11+0+1101
1101101	10+1+0+1
10101	

### Cách tính điểm

Với mỗi xâu truy vấn, gọi  $N$  là độ dài xâu,  $K$  là số bit 1 trong một xâu truy vấn đó.

- Subtask 1 (15 điểm):  $N \leq 16$ .
- Subtask 2 (30 điểm):  $N \leq 10^5$ ,  $K \leq 10^3$ .
- Subtask 3 (55 điểm):  $N \leq 10^5$ .

Với mỗi subtask, gồm nhiều test. Với mỗi test, gọi  $out$  là tổng số bit 1 của tất cả các truy vấn tính được theo cách phân tích của chương trình thí sinh.  $ans$  là tổng số bit 1 của tất cả các truy vấn tính được theo cách phân tích của ban giám khảo. Điểm của thí sinh nhận được cho test đó được tính bằng  $(\frac{ans}{out})^2$ . Điểm của mỗi subtask được tính bằng điểm thấp nhất của một test trong subtask đó.