

Đề Tin_11
(Đề thi gồm 6 trang)

KỲ THI CHỌN HỌC SINH GIỎI
LẦN THỨ XVII NĂM 2026
ĐỀ THI MÔN: TIN HỌC – LỚP 11
Thời gian làm bài: 180 phút
(Không kể thời gian giao đề)

TỔNG QUAN ĐỀ THI

Tên bài	Tên chương trình	Dữ liệu đầu vào	Kết quả đầu ra	Điểm
Nướng thịt	tomahawk.*	tomahawk.inp	tomahawk.out	6
Thang máy siêu hẹp	dizalo.*	dizalo.inp	dizalo.out	7
Vương quốc cổ đại	gstree.*	gstree.inp	gstree.out	7

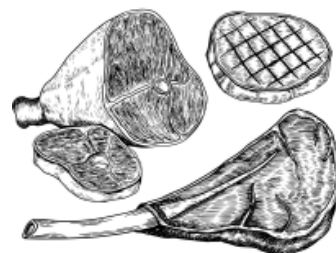
Dấu * được thay thế bởi py hoặc cpp của ngôn ngữ lập trình tương ứng là Python và C++

Hãy lập trình giải các bài toán sau:

Bài 1. Nướng thịt

Người đầu bếp tại nhà hàng nổi tiếng nhất ở đất cổ đô đang chuẩn bị những miếng thịt bò Kobe cho một nhóm lập trình viên trẻ tuổi, vui vẻ và hiện đang rất đói.

Miếng thịt có thể được biểu diễn dưới dạng một ma trận $n \times n$. Ban đầu, nhiệt độ của toàn bộ miếng thịt là 0 độ. Trong quá trình nấu, nhiệt độ ở các phần khác nhau của miếng thịt (tức là các ô khác nhau trong ma trận) sẽ tăng lên.



Để chế biến miếng thịt Kobe hoàn hảo, đầu bếp nhấc miếng thịt lên và đặt lại lên vỉ nướng q lần. Mỗi lần, đầu bếp có thể đặt miếng thịt theo một trong ba cạnh: trái, phải hoặc dưới. Sau đó, ông sẽ để miếng thịt nướng ở cạnh đó trong x giây.

Khi miếng thịt được nướng ở cạnh dưới, nó có thể được nướng tối đa n giây. Sau đó, nhiệt độ của tất cả các ô ở hàng thứ n trong ma trận sẽ tăng thêm x , nhiệt độ của các ô ở hàng thứ $n - 1$ sẽ tăng thêm $x - 1$, nhiệt độ của các ô ở hàng thứ $n - 2$ sẽ tăng thêm $x - 2$, ..., nhiệt độ của các ô ở hàng thứ $n - x + 1$ sẽ tăng thêm 1.

Khi miếng thịt được nướng ở cạnh phải, nó có thể được nướng tối đa $\left\lfloor \frac{n+1}{2} \right\rfloor$ giây. Sau đó, nhiệt độ của tất cả các ô ở cột thứ n trong ma trận sẽ tăng thêm x , nhiệt độ của các ô ở cột thứ $n - 1$ sẽ tăng thêm $x - 1$, nhiệt độ của các ô ở cột thứ $n - 2$ sẽ tăng thêm $x - 2$, ..., nhiệt độ của các ô ở cột thứ $n - x + 1$ sẽ tăng thêm 1.

Tương tự, khi miếng thịt được nướng ở cạnh trái, nó có thể được nướng tối đa $\left\lfloor \frac{n+1}{2} \right\rfloor$ giây. Sau đó, nhiệt độ của tất cả các ô ở cột thứ nhất trong ma trận sẽ tăng thêm x , nhiệt độ của các ô ở cột thứ hai sẽ tăng thêm $x - 1$, nhiệt độ của các ô ở cột thứ ba sẽ tăng thêm $x - 2$, ..., nhiệt độ của các ô ở cột thứ x sẽ tăng thêm 1.

Chúng ta biết rằng độ ngon của miếng bò Kobe có liên quan chặt chẽ đến sự chênh lệch nhiệt độ, vì vậy đầu bếp muốn tìm hiệu giữa phần lạnh nhất và phần nóng nhất của miếng thịt!

Dữ liệu vào (Input)

- Dòng đầu tiên chứa hai số tự nhiên n và q ($1 \leq n \leq 10^9, 1 \leq q \leq 10^5$) lần lượt là kích thước của miếng thịt và số lần đầu bếp nhắc miếng thịt.
- Trong mỗi dòng trong q dòng tiếp theo, có một ký tự s và một số tự nhiên x . Ký tự s chỉ cạnh của miếng thịt bò Kobe: "L" – trái, "R" – phải, "D" – dưới.

Nếu $s = "D"$ thì $1 \leq x \leq n$. Ngược lại, $1 \leq x \leq \left\lfloor \frac{n+1}{2} \right\rfloor$.

Kết quả (Output)

In ra một số duy nhất – là hiệu (đơn vị độ) giữa ô có nhiệt độ thấp nhất và ô có nhiệt độ cao nhất trên miếng thịt

Ví dụ:

Input	Output
4 2 L 2 R 1	2
3 3 R 2 D 3 R 2	6

Giải thích các bước tăng nhiệt độ trong 2 ví dụ

Ví dụ 1

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

Ví dụ 2

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 5 & 7 \end{bmatrix}$$

Ràng buộc

- Subtask 1: 10% ứng với $n, q \leq 20$
- Subtask 2: 30% ứng với $n, q \leq 1000$
- Subtask 3: 30% ứng với $n \leq 1000$
- Subtask 4: 20% ứng với n là số chẵn
- Subtask 5: 10% không có ràng buộc gì thêm

Câu 2: Thang máy siêu hẹp

Trong một thành phố nọ có một tòa nhà cao n tầng. Có n người đang chờ thang máy ở tầng trệt. Người thứ i muốn lên tầng a_i . Không có hai người nào muốn lên cùng một tầng. Tòa nhà có một thang máy đủ rộng cho tất cả mọi người, nhưng nó quá hẹp đến nỗi hai người không thể đứng cạnh nhau; người này phải đứng sau người kia.

Mọi người đều vào thang máy, nhưng họ không nghĩ đến thứ tự ra khỏi thang máy! Ban đầu, người thứ i ở vị trí i , nhìn từ cửa thang máy. Nếu một người muốn ra khỏi thang máy, tất cả những người phía trước họ (gần cửa hơn) phải tạm thời ra khỏi thang máy. Khi quay trở lại thang máy, họ có thể sắp xếp lại thứ tự theo ý muốn. Những người ở phía sau (xa cửa hơn) người muốn ra khỏi thang máy sẽ không ra khỏi thang máy.



Hình minh họa ở trên cho thấy thứ tự xuất phát của mọi người trong thang máy ở lượt đầu tiên. Thang máy đang ở tầng 1, và người ở vị trí số 3 muốn ra ngoài. Để họ ra ngoài, những người ở vị trí số 1 và số 2 cũng phải ra ngoài.

Mirko đang quan sát tình hình hiện tại và suy nghĩ. Anh ấy muốn biết sẽ có bao nhiêu lượt ra khỏi thang máy nếu mọi người luôn quay trở lại thang máy một cách tối ưu (Để tối ưu, họ sẽ sắp xếp sao cho số lần người phải bước ra ở các tầng sau là ít nhất). Nếu một người ra khỏi thang máy nhiều lần, mỗi lần được tính riêng. Slavko – người bạn của Mirko cũng đưa cho anh ấy câu hỏi: Nếu người ở vị trí x_i không ở trong thang máy, thì sẽ có bao nhiêu lượt người đi ra khỏi thang máy?

Mirko sẽ phải trả lời cho câu hỏi của mình và các câu hỏi của Slavko. Lưu ý rằng đối với mỗi câu hỏi, tất cả những người không ở trong thang máy từ các câu hỏi trước cũng sẽ không ở trong thang máy hiện tại. Bạn hãy giúp Mirko giải quyết vấn đề này!

Lưu ý: Thang máy sẽ luôn di chuyển từ tầng 1 đến tầng n và dừng lại ở mọi tầng mà có người muốn ra.

Dữ liệu vào(input):

- Dòng đầu tiên chứa hai số nguyên không âm n và q ($0 \leq q < n \leq 10^5$), số người/tầng và số câu hỏi.
- Dòng thứ hai chứa n số nguyên a_i ($1 \leq a_i \leq n$, $a_i \neq a_j$ với mỗi $i \neq j$), trong đó a_i là tầng mà người thứ i muốn ra khỏi thang máy. Dãy (a_i) là một hoán vị.
- Dòng thứ ba chứa q số nguyên x_i ($1 \leq x_i \leq n$, $x_i \neq x_j$ với mỗi $i \neq j$), lần lượt là vị trí của các người không ở trong thang máy trong câu hỏi của Slavko.

Dữ liệu ra(output):

- Trong một dòng, hãy in ra $q + 1$ số trả lời cho $q+1$ trường hợp: ban đầu và sau mỗi lần có một người ở vị trí x_i không đi thang máy nữa.

Ví dụ:

input	output
5 2	9 6 4
3 4 1 2 5	
3 2	

Giải thích:

Trường hợp ban đầu:

Để tối ưu, khi những người tạm bước ra quay trở lại, họ sẽ đứng theo thứ tự: ai xuống tầng cao hơn thì đứng xa cửa hơn.

- Tầng 1: Người muốn xuống tầng 1 đang đứng ở vị trí 3.
 - Để người này ra, người ở vị trí 1 và 2 phải bước ra.
 - Tổng lượt ra: 3 (vị trí 1, 2 và người tầng 1).
 - Còn lại trong thang: {3, 4, 2, 5}. Họ sắp xếp lại để người tầng 2 đứng gần cửa nhất.
- Tầng 2: Người muốn xuống tầng 2 hiện đang ở gần cửa.
 - 2 người đầu hàng vẫn phải bước ra để người tầng 2 bước ra.
 - Tổng lượt ra: 3
 - Còn lại: {3, 4, 5}.
- Tầng 3: Người muốn xuống tầng 3 đang ở gần cửa nhất, và họ bước ra luôn. Vậy chỉ cần 1 lượt bước ra.

- Tầng 4: Người muốn xuống tầng 4 đang ở gần cửa nhất. Vậy chỉ cần 1 người bước ra.
- Tầng 5: Người cuối cùng: Chỉ cần 1 người bước ra.

Tổng cộng: $3 + 3 + 1 + 1 + 1 = 9$ lượt ra.

Với 2 câu hỏi sau:

1. Loại bỏ người ở vị trí 3 (người xuống tầng 1):
 - Trong thang còn các người muốn xuống tầng: $\{3, 4, 2, 5\}$.
 - Tính toán tương tự, tổng lượt ra lúc này giảm xuống còn 6.
2. Loại bỏ tiếp người ở vị trí 2 (người xuống tầng 4):
 - Trong thang còn: $\{3, 2, 5\}$.
 - Tổng lượt ra lúc này chỉ còn 4.

Kết quả cuối cùng in ra: 9 6 4.

Ràng buộc:

- Sub 1: 16% điểm ứng với $n, q \leq 100$
- Sub 2: 19% điểm ứng với $n, q \leq 1\,000$
- Sub 3: 29% điểm ứng với $q = 0$
- Sub 4: 46% điểm không có ràng buộc gì thêm

Bài 3. Vương quốc cổ đại

Trong một vương quốc cổ đại, có n thành phố được đánh số từ 1 đến n . Mỗi thành phố mang một “mật mã cổ” là một số nguyên dương a_i , đại diện cho đặc trưng văn hóa và lịch sử riêng của thành phố đó.

Nhà vua muốn xây dựng một hệ thống đường đi để kết nối tất cả các thành phố lại với nhau, nhằm tăng cường giao thương và phát triển toàn vương quốc. Tuy nhiên, việc xây dựng đường giữa hai thành phố không đơn giản, vì chi phí phụ thuộc vào mức độ “tương đồng” giữa hai nơi.

Cụ thể:

- Mức độ tương đồng giữa hai thành phố được đo bằng **ước số chung lớn nhất** của hai mật mã, tức là $GCD(a_i, a_j)$.
- Chi phí xây dựng con đường nối giữa hai thành phố i và j được xác định bởi:

$$123456 - GCD(a_i, a_j)$$

Điều này có nghĩa là:

- Hai thành phố càng có nhiều điểm chung (GCD lớn) thì càng dễ kết nối (chi phí thấp),

- Ngược lại, hai thành phố càng khác biệt thì chi phí xây dựng càng cao.

Yêu cầu

Nhà vua muốn chọn ra một hệ thống đường sao cho:

- Tất cả các thành phố đều được kết nối (trực tiếp hoặc gián tiếp),
- Tổng chi phí xây dựng là **nhỏ nhất có thể**.

Nhiệm vụ của bạn là tính tổng chi phí nhỏ nhất đó.

Dữ liệu vào(input):

- Dòng đầu chứa số nguyên n — số lượng thành phố.
- Dòng thứ hai chứa n số nguyên dương $a_1, a_2, \dots, a_n$ ($a_i \leq 10^5$).

Dữ liệu ra(output):

- In ra một số duy nhất là **tổng chi phí nhỏ nhất** để kết nối tất cả các thành phố.

Ví dụ

input	output
3 10 20 30	246892

Ràng buộc:

- Subtask 1: $n \leq 500$
- Subtask 2: $n \leq 50000$

----- HẾT -----