

## Mục lục

<b>Contest #8</b>	<b>1</b>
Trò chơi đồ thị — GRAPHGAME . . . . .	2
Đoạn bít liên tiếp – CONSBIT . . . . .	3
Cây nho – GRAPE . . . . .	4
<b>Contest #7</b>	<b>6</b>
Yến tiệc . . . . .	6
Bãi đậu xe đạp – BIKE . . . . .	8
Thủ thư . . . . .	10
<b>Contest #6</b>	<b>12</b>
Nghịch thế — INVERSION . . . . .	12
Khối lập phương . . . . .	13
Sắc màu . . . . .	15
<b>Contest #5</b>	<b>17</b>
Kim tự tháp 2 — MEDPYRH . . . . .	17
HNTOUR - Khám phá Hà Nội . . . . .	19
Pháo hoa . . . . .	21
<b>Contest #4</b>	<b>23</b>
BITPUZZLE - Cầu đổ . . . . .	23
Nhiên liệu . . . . .	25
SUMMERCAMP - Trại hè . . . . .	27
<b>Contest #3</b>	<b>29</b>
Trung tâm hội nghị . . . . .	29
BALLOON . . . . .	31
CODING . . . . .	32
<b>Contest #2</b>	<b>35</b>
SPIDERMAN . . . . .	35
Diệt Nấm . . . . .	36
Du lịch Hạ Long . . . . .	37
<b>Contest #1</b>	<b>39</b>
GEMSTONE . . . . .	39
ESCAPING . . . . .	41
LANDSLIDE . . . . .	43

Nộp bài tại: <https://northern.contest.codeforces.com>

## Bài 1. Trò chơi đồ thị

Khi học lý thuyết đồ thị, Ninh và Giang cùng nhau chơi một trò chơi trên đồ thị. Đầu tiên Ninh có một đơn đồ thị vô hướng  $G$  có  $N$  đỉnh được đánh số từ 1 đến  $N$ . Tiếp theo Ninh tiếp tục ghi ra giấy danh sách các đỉnh kề của từng đỉnh. Sau đó Giang thay đổi danh sách các đỉnh kề của 1 hoặc 2 đỉnh bằng cách thay đổi một số đỉnh kề với chúng. Cụ thể, nếu một đỉnh ban đầu có  $X$  đỉnh kề, Giang sẽ ghi ra đủ  $X$  đỉnh kề mà một số đỉnh trong đó khác với danh sách đỉnh kề ban đầu.

**Yêu cầu:** Cho biết số lượng đỉnh mà Giang đã thay đổi và danh sách các đỉnh kề của từng đỉnh sau khi đã thay đổi, hãy giúp Ninh tìm ra các đỉnh kề bị Giang thay đổi.

### Dữ liệu vào

- Dòng thứ nhất chứa một số nguyên duy nhất  $P \in \{1, 2\}$  là số lượng đỉnh mà Giang đã thay đổi danh sách các đỉnh kề của chúng.
- Dòng thứ hai ghi duy nhất một số nguyên dương  $N$  ( $3 \leq N \leq 10^5$ ) là số lượng đỉnh của đồ thị  $G$ .
- Dòng thứ  $i$  trong số  $N$  dòng tiếp theo mô tả danh sách kề của đỉnh  $i$  đã bị Giang thay đổi:
  - đầu tiên là một số nguyên dương  $K_i \leq N - 1$ , là số lượng đỉnh kề của đỉnh  $i$ ;
  - $K_i$  số nguyên tiếp theo là các đỉnh kề của đỉnh  $i$ .
- Biết rằng  $K_1 + K_2 + \dots + K_N \leq 4 \times 10^5$ .

Các số trên cùng một dòng cách nhau bởi dấu cách.

### Kết quả

- Nếu  $P = 1$ , ghi ra duy nhất một đỉnh mà Giang đã thay đổi.
- Nếu  $P = 2$ , ghi ra trên cùng dòng 2 đỉnh mà Giang đã thay đổi theo thứ tự tăng dần.

Dữ liệu đảm bảo luôn có lời giải và có duy nhất một lời giải.

### Ví dụ

test	answer
2 7 4 7 3 2 4 2 6 1 4 7 6 4 1 4 1 3 5 6 2 4 1 3 7 5 1 1 3	1 6

### Hạn chế

- Subtask 1: Có 40% số test có  $P = 1$ .
- Subtask 2: Không có ràng buộc gì thêm.

## Bài 2. Đoạn bit liên tiếp

File dữ liệu vào: `standard input`  
 File kết quả: `standard output`  
 Hạn chế thời gian: 3 seconds  
 Hạn chế bộ nhớ: 1024 megabytes

Cho một dãy bit gồm  $n$  bit, đồng thời cho  $q$  truy vấn mà mỗi truy vấn thuộc một trong hai dạng:

- 1  $i$   $c$ : đặt bit thứ  $i$  bằng  $c$  ( $1 \leq i \leq n$ ,  $c \in \{0, 1\}$ ).
- 2  $l$   $r$   $x$   $y$ : tìm hai số nguyên  $p, q$  thỏa mãn  $l \leq p \leq q \leq r$  và đoạn bit từ bit thứ  $p$  đến bit thứ  $q$  thỏa mãn chứa đoạn bit 0 liên tiếp dài nhất có độ dài đúng bằng  $x$  và đoạn bit 1 liên tiếp dài nhất có độ dài đúng bằng  $y$ . Dữ liệu đảm bảo ( $1 \leq l \leq r \leq n$ ,  $0 \leq x, y \leq n$ ).

### Dữ liệu vào

Dòng đầu là hai số nguyên  $n$  và  $q$  ( $1 \leq n, q \leq 2 \times 10^5$ ).

Dòng thứ hai là một dãy bit dài  $n$  bit.

Dòng thứ  $i$  trong số  $q$  dòng tiếp theo mô tả một truy vấn, mỗi truy vấn thuộc một trong hai dạng như đã nêu ở trên.

### Kết quả

In ra  $q$  dòng, là kết quả của truy vấn tương ứng.

- Nếu không tồn tại cặp số  $(p, q)$  thỏa mãn truy vấn, in ra  $-1$ .
- Ngược lại, in ra một cặp số  $(p, q)$  bất kỳ thỏa mãn truy vấn.

### Ví dụ

standard input	standard output
10 7	6 8
0011100101	6 8
2 1 10 2 1	-1
2 4 8 2 1	-1
1 1 1	9 10
2 1 5 2 1	
2 6 7 0 0	
1 6 0	
2 5 10 1 1	

### Hạn chế

- (Subtask 1) 10% số test ứng với 10% số điểm thỏa mãn:  $1 \leq n, q \leq 100$ .
- (Subtask 2) 20% số test ứng với 20% số điểm thỏa mãn:  $1 \leq n, q \leq 5000$ .
- (Subtask 3) 30% số test ứng với 30% số điểm thỏa mãn: Sẽ chỉ có đúng một truy vấn loại 2.
- (Subtask 4) 40% số test ứng với 40% số điểm còn lại không có ràng buộc gì thêm.

## Bài 3. Cây nho – GRAPE

File dữ liệu vào: `stdin`  
 File kết quả: `stdout`  
 Hạn chế thời gian: 1 giây  
 Hạn chế bộ nhớ: 512 MB

Cây Nho có  $N$  khớp lá, được đánh số từ 1 đến  $N$ , các khớp lá kết nối với nhau bởi  $N - 1$  nhánh. Nhánh  $i$  nối hai khớp  $A_i$  và  $B_i$  với độ dài là  $W_i$ . Mỗi khớp đều kết nối trực tiếp hoặc gián tiếp đến mọi khớp khác.

Sơn có thể thực hiện ba loại thao tác trên Cây Nho:

- **Tìm kiếm:** Từ một khớp cho trước, tìm khoảng cách đến quả nho gần nhất. Nếu không có quả nho nào, trả về  $-1$ .
- **Tưới:** Tưới vào một khớp, khiến nó mọc ra một quả nho hoặc làm rơi quả nho ở đó.
- **Sửa đổi:** Thay đổi chiều dài của một nhánh cụ thể giữa hai khớp đến một giá trị mới.

Cho  $Q$  thao tác được lên kế hoạch, tìm khoảng cách đến quả nho gần nhất cho mỗi thao tác tìm kiếm.

### Dữ liệu vào

- Dòng đầu: Hai số nguyên  $N$  và  $Q$ .
- $N - 1$  dòng tiếp theo: Mỗi dòng ba số nguyên  $A_i$ ,  $B_i$ ,  $W_i$ , đại diện cho một nhánh.
- $Q$  dòng tiếp theo: Mỗi dòng mô tả một thao tác:
  - 1  $q_i$ : Thao tác tìm kiếm từ khớp  $q_i$ .
  - 2  $u_i$ : Thao tác tưới vào khớp  $u_i$ .
  - 3  $a_i$   $b_i$   $w_i$ : Thao tác sửa đổi chiều dài nhánh giữa  $a_i$  và  $b_i$  thành  $w_i$ .

### Kết quả

Với mỗi thao tác tìm kiếm, in ra khoảng cách đến quả nho gần nhất hoặc  $-1$  nếu không có quả nho nào.

### Hạn chế

- $2 \leq N \leq 100000$
- $1 \leq Q \leq 100000$
- $1 \leq A_i \neq B_i \leq N$
- $0 \leq W_i \leq 10^9$

Subtask	Điểm	Hạn chế bổ sung
1	15	$N, Q \leq 2000$
2	15	Đối với tất cả thao tác tìm kiếm, $q_i = 1$
3	15	Cây nho có dạng cây nhị phân đầy đủ, $A_i = \lfloor \frac{i+1}{2} \rfloor$ , $B_i = i + 1$
4	15	Tại mọi thời điểm, có tối đa 1 quả nho trên cây nho
5	15	Tất cả thao tác tưới sẽ thực hiện trước thao tác tìm kiếm hoặc sửa đổi. Đối với mọi thao tác sửa đổi, $w_i = 0$
6	25	Không có ràng buộc bổ sung

**Ví dụ**

stdin	stdout
6 11	7
1 2 3	2
1 3 4	-1
2 4 1	4
2 5 4	
3 6 6	
2 3	
1 2	
2 4	
3 1 3 2	
1 1	
2 3	
3 2 1 2	
2 4	
1 3	
2 2	
1 3	

## Bài 4. Yến tiệc

File dữ liệu vào: BANQUET.INP  
 File kết quả: BANQUET.OUT  
 Hạn chế thời gian: 1 second  
 Hạn chế bộ nhớ: 256 megabytes

Hoàng gia Lục đang dự định tổ chức một buổi yến tiệc cho các nhà vua của các vương quốc kết nghĩa. Lục Vương dự kiến sắp xếp một bàn yến tiệc dài gồm  $N$  món ăn xếp thành một hàng thẳng, đánh số từ 1 đến  $N$ .

Một món ăn có thể ngon nhưng không hợp khẩu vị với khách mời của buổi yến tiệc, vì vậy, đầu bếp đã gán cho món ăn thứ  $i$  một giá trị  $A_i$  là độ hợp khẩu vị của món ăn đối với khách mời. Nếu món thứ  $i$  có thể khiến khách mời không hài lòng, nó sẽ mang giá trị  $A_i$  không dương.

Có tất cả  $K$  nhà vua được gửi lời mời, mỗi người sẽ được thưởng thức một dãy các món ăn liên tiếp được xếp trên bàn. Dãy thức ăn này có thể là một dãy rỗng. Dãy thức ăn của các nhà vua khác nhau không được giao nhau, vì mỗi món không thể ăn hai lần.

Hoàng gia Lục muốn chọn các dãy thức ăn cho các nhà vua sao cho tổng các độ hài lòng của tất cả các món ăn được thưởng thức là lớn nhất.

### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên  $N, K$  ( $1 \leq K \leq N \leq 3 \times 10^5$ );
- Dòng tiếp theo chứa  $N$  số nguyên  $A_1, A_2, \dots, A_N$  ( $|A_i| \leq 10^9$ ).

### Kết quả

- In ra kết quả bài toán.

### Ví dụ

BANQUET.INP	BANQUET.OUT
6 1 1 -2 3 -1 5 -6	7
6 2 1 2 3 -10 5 6	17
6 4 -1 -2 -1 0 -5 -1	0

### Lưu ý

- Trong ví dụ thứ 1, chọn dãy  $[3, -1, 5]$  cho nhà vua duy nhất.
- Trong ví dụ thứ 2, chọn hai dãy  $[1, 2, 3]$  và  $[5, 6]$ .
- Trong ví dụ thứ 3, do các món ăn có khả năng không hợp khẩu vị với các nhà vua khác nên chọn dãy rỗng cho cả 4 nhà vua.

### Hạn chế

- Subtask 1 (10%):  $A_i \geq 0$ .
- Subtask 2 (10%): Có nhiều nhất một giá trị  $A_i < 0$ .
- Subtask 3 (10%):  $K = 1$ .
- Subtask 4 (10%):  $1 \leq K \leq N \leq 80$ .
- Subtask 5 (20%):  $1 \leq K \leq N \leq 300$ .
- Subtask 6 (20%):  $1 \leq K \leq N \leq 2000$ .
- Subtask 7 (20%): Không có giới hạn gì thêm.



## Bài 5. Bãi đậu xe đạp – BIKE

File dữ liệu vào:	<code>stdin</code>
File kết quả:	<code>stdout</code>
Hạn chế thời gian:	1 giây
Hạn chế bộ nhớ:	512 MB

Có  $N$  khu vực trong bãi đậu xe, được đánh số từ 0 (khu vực tốt nhất) đến  $N - 1$  (khu vực tệ nhất). Khu vực thứ  $i$  có  $x_i$  chỗ đậu xe và  $y_i$  người dùng nhắm tới khu vực đó. Có đủ chỗ cho tất cả người dùng nói chung ( $\sum_i y_i \leq \sum_i x_i$ ), nhưng không nhất thiết đúng khu vực mục tiêu. Chúng ta cần chỉ định chính xác một chỗ cho mỗi người dùng.

Điểm số của bạn được định nghĩa bởi  $U - D$ , trong đó  $U$  là tổng số người dùng được chỉ định vào khu vực thấp hơn so với mục tiêu của họ và  $D$  là tổng số người dùng được chỉ định vào khu vực cao hơn so với mục tiêu của họ.

Nói cách khác, bạn được  $+1$  khi chỉ định vào khu vực thấp hơn mục tiêu,  $0$  khi chỉ định vào khu vực mục tiêu và  $-1$  khi chỉ định vào khu vực cao hơn mục tiêu.

**Yêu cầu:** Xác định điểm tối đa có thể là bao nhiêu?

### Dữ liệu vào

- Dòng đầu tiên chứa một số nguyên  $N$ , số khu vực hoặc mức đăng ký.
- Dòng thứ hai chứa  $N$  số nguyên  $x_0, x_1, \dots, x_{N-1}$ , số lượng chỗ đậu trong mỗi khu vực.
- Dòng thứ ba chứa  $N$  số nguyên  $y_0, y_1, \dots, y_{N-1}$ , số lượng người dùng có mỗi mức đăng ký.

### Kết quả

Ghi một số nguyên duy nhất, giá trị tối đa của  $U - D$  bằng cách phân bổ người dùng vào các chỗ đậu xe một cách tối ưu.

### Hạn chế

- $1 \leq N \leq 3 \cdot 10^5$ .
- $0 \leq x_i, y_i \leq 10^9$  với  $i = 0, 1, \dots, N - 1$ .
- $y_0 + y_1 + \dots + y_{N-1} \leq x_0 + x_1 + \dots + x_{N-1} \leq 10^9$ .

### Các Subtask

- Subtask 1 (20 điểm):  $N = 2$ ,  $x_i \leq 100$ ,  $y_i \leq 100$ .
- Subtask 2 (8 điểm):  $x_i = x_j = y_i = y_j$  cho mọi  $i, j$ . Nói cách khác, tất cả các giá trị  $x$  và  $y$  trong đầu vào đều bằng nhau.
- Subtask 3 (14 điểm):  $x_i, y_i \leq 1$ .
- Subtask 4 (36 điểm):  $N, x_i, y_i \leq 100$ .
- Subtask 5 (22 điểm): Không có ràng buộc bổ sung.

**Ví dụ**

stdin	stdout
3 2 3 1 1 2 1	2

**Giải thích**

Trong ví dụ này, có 3 khu vực đỗ xe với số lượng chỗ đỗ lần lượt là 2, 3, và 1. Có 3 mức đăng ký với số lượng người dùng cho từng mức lần lượt là 1, 2, và 1.

Ta có thể sắp xếp như sau để tối ưu hóa đánh giá:

- Người dùng mức 0 được gán vào khu vực 0 (tương ứng), dẫn đến không có đánh giá (trung tính).
- Người dùng mức 1 được gán vào hai chỗ trong khu vực 0, dẫn đến 2 đánh giá tích cực (vì khu vực  $0 < 1$ ).
- Người dùng mức 2 được gán vào khu vực 1, dẫn đến không có đánh giá (trung tính).

Tổng số đánh giá tích cực là 2 và không có đánh giá tiêu cực nào, nên kết quả là  $2 - 0 = 2$ .

## Bài 6. Thủ thư

File dữ liệu vào:	LIBRARIAN.INP
File kết quả:	LIBRARIAN.OUT
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	512 megabytes

An trong khi muốn đi làm thêm đã chọn làm thủ thư tại một thư viện trong thành phố. Tại đây có  $N$  kệ để sách, mỗi kệ có  $M$  ngăn chứa sách. Thư viện có tất cả  $K$  quyển sách được đánh số từ 1 đến  $K$  (và tất nhiên có thể xếp đủ  $K$  quyển này lên kệ). Ngăn sách thứ  $j$  của kệ sách thứ  $i$  gọi là ngăn  $(i, j)$ , được biểu diễn bởi một số nguyên  $a_{i,j}$ . Nếu  $a_{i,j} = 0$ , ngăn sách đang trống, ngược lại, nó đang chứa quyển sách mang chỉ số tương ứng.

Hiện tại sách trên các kệ đang bị thay đổi vị trí quá nhiều và cậu ta muốn sắp xếp lại các quyển sách như ban đầu. Ban đầu ngăn thứ  $j$  của kệ sách thứ  $i$  đang chứa quyển sách mang chỉ số  $b_{i,j}$ . Nếu  $b_{i,j} = 0$ , ngăn sách này trống. Cậu có thể sắp xếp bằng hai thao tác sau:

- Chuyển một quyển sách sang ngăn phía bên trái hoặc bên phải nó nếu ngăn bên trái hoặc bên phải đó đang trống.
- Cầm một quyển sách và chuyển sang một ngăn đang trống trên một kệ bất kì.

Là một người cẩn thận, An chỉ chuyển một quyển sách một lúc.

Nhưng do hôm qua, cậu đã phải lau dọn lại cả thư viện nên cột sống của cậu hôm nay cũng không còn ổn, vì vậy cậu muốn thực hiện thao tác loại 2 ít nhất có thể. Hãy giúp An tính được số lượng thao tác đó là bao nhiêu để cậu có thể mua thuốc cho phù hợp?

### Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương  $N, M$  ( $1 \leq N, M \leq 10^3$ );
- $N$  dòng tiếp theo, mỗi dòng chứa  $M$  số nguyên  $a_{i,j}$  ( $0 \leq a_{i,j} \leq K$ );
- $N$  dòng tiếp theo, mỗi dòng chứa  $M$  số nguyên  $b_{i,j}$  ( $0 \leq b_{i,j} \leq K$ ).

### Kết quả

- In ra kết quả bài toán, nếu không thể sắp xếp, in ra  $-1$ .

### Hạn chế

- Subtask 1 (20%):  $M \leq 100$ ; các quyển sách ở cùng kệ sách trong trạng thái ban đầu.
- Subtask 2 (40%): Các quyển sách ở cùng kệ sách trong trạng thái ban đầu.
- Subtask 3 (40%): Không có giới hạn gì thêm.

**Ví dụ**

LIBRARIAN.INP	LIBRARIAN.OUT
2 4 1 0 2 0 3 5 4 0 2 1 0 0 3 0 4 5	2
3 3 1 2 3 4 5 6 7 8 0 4 2 3 6 5 1 0 7 8	4
2 2 1 2 3 4 2 3 4 1	-1

**Lưu ý**

Ở ví dụ thứ nhất, An sẽ chuyển quyển sách 1 sang phải. Cầm quyển 2 và đặt vào ngăn sách thứ nhất của kệ sách thứ nhất. Cuối cùng, cậu cầm quyển 5 và đặt vào ngăn sách thứ tư của kệ sách thứ 2.

## Bài 7. Nghịch thế

Số lượng nghịch thế của một dãy số  $a_1, a_2, \dots, a_n$  là số lượng cặp  $(a_i, a_j)$  thỏa mãn  $i < j$  và  $a_i > a_j$ .

Yêu cầu: Cho số  $n$  và số lượng nghịch thế  $m$ , hãy tìm hoán vị nhỏ nhất của tập  $\{1, 2, \dots, n\}$  sao cho số lượng nghịch thế của nó đúng bằng  $m$ .

Nhắc lại: một hoán vị  $a_1, a_2, \dots, a_n$  gọi là nhỏ hơn hoán vị  $b_1, b_2, \dots, b_n$  khi và chỉ khi tồn tại một số nguyên  $k$  sao cho  $a_j = b_j$  với  $1 \leq j < k$  và  $a_k < b_k$ .

### Dữ liệu vào

Dòng đầu tiên ghi một số  $T$  ( $T \leq 50$ ) là số lượng bộ test.

Dòng thứ  $i$  trong số  $T$  dòng tiếp theo chứa hai số nguyên  $n$  và  $m$  ( $m \leq 10^{12}$ ).

### Kết quả

Ghi ra  $T$  dòng, mỗi dòng một số  $q$  là giá trị duy nhất được tính theo hoán vị  $a_1, a_2, \dots, a_n$  tìm được tương ứng với dòng trong dữ liệu đầu vào. Số  $q$  là phần dư của phép chia công thức sau cho  $10^9 + 7$ :

$$a_1 + a_2 * 2^1 + a_3 * 2^2 + \dots + a_n * 2^{n-1}.$$

Nếu không tồn tại hoán vị thỏa mãn thì ghi ra -1.

### Ví dụ

test	answer
2	58
5 9	673
7 3	

### Giải thích

- $n = 5, m = 9$ , hoán vị tìm được: 4 5 3 2 1  $\rightarrow q = 58$ ;
- $n = 7, m = 3$ , hoán vị tìm được: 1 2 3 4 7 6 5  $\rightarrow q = 673$ .

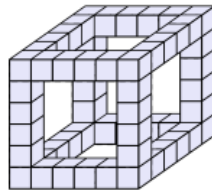
### Hạn chế

- Subtask 1 (25%):  $1 \leq n \leq 10$ ;
- Subtask 2 (25%):  $1 \leq n \leq 1000$ ;
- Subtask 3 (25%):  $1 \leq n \leq 10^5$ ;
- Subtask 4 (25%):  $1 \leq n \leq 10^6$ .

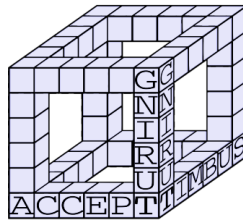
## Bài 8. Khối lập phương

File dữ liệu vào:	CUBIK.INP
File kết quả:	CUBIK.OUT
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	1024 megabytes

Cubik là một loại câu đố đặc biệt kết hợp dựa trên hình khối lập phương và các câu từ. Khi xây dựng một cubik, bạn bắt đầu bằng việc chọn một số nguyên dương  $a$  : chiều dài cạnh của khối lập phương. Sau đó, bạn xây dựng một khối lập phương lớn gồm  $a \times a \times a$  khối lập phương nhỏ. Khối lập phương lớn này có 12 cạnh, ta có thể đánh số các cạnh từ 1 đến 12. Sau đó, bạn loại bỏ tất cả các khối lập phương nhỏ không tiếp xúc với các cạnh của khối lập phương lớn. Hình vẽ dưới đây cho thấy vật thể mà bạn sẽ thu được khi  $a = 6$ .



Cuối cùng, bạn gán một chữ cái cho mỗi khối lập phương nhỏ trong vật thể đó. Bạn phải tạo ra một từ có ý nghĩa trên mỗi cạnh của khối lập phương lớn. Mỗi cạnh có thể đọc theo cả hai hướng và chỉ cần một trong hai hướng đọc phải tạo ra một từ có ý nghĩa. Hình dưới đây cho thấy vật thể với  $a = 6$  trong đó một số khối lập phương nhỏ đã được gán chữ cái. Bạn đã có thể đọc các từ "SUBMIT", "ACCEPT" và "TURING" trên ba cạnh của khối lập phương lớn.



Bạn được cho một danh sách gồm  $N$  từ hợp lệ. Mỗi từ trong danh sách từ có thể xuất hiện trên nhiều cạnh của một cubik hợp lệ. Hãy xác định số lượng các cubik hợp lệ khác nhau có thể được xây dựng, lấy phần dư chia 998244353. Hai cubik được xem là khác nhau nếu như đặt 2 cubik gần nhau sao cho mọi cặp cạnh được đánh cùng số song song với nhau thì tồn tại ít nhất 1 cặp cạnh mà từ được viết trên chúng là khác nhau (nếu cùng 1 từ nhưng lại viết ngược nhau thì cũng tính là khác nhau).

### Dữ liệu vào

- Dòng đầu tiên chứa một số nguyên  $N (1 \leq N \leq 10^5)$  là số lượng từ.
  - Mỗi dòng trong số  $N$  dòng tiếp theo chứa một từ  $S_i (3 \leq |S_i| \leq 10)$ .
- Dữ liệu đảm bảo tất cả đều khác nhau.

### Kết quả

- In ra kết quả bài toán, chia lấy dư cho 998244353.

## Hạn chế

- Subtask 1 (25%): Tất cả các từ gồm các kí tự  $'a' \rightarrow 'f'$
- Subtask 2 (25%): Tất cả các từ gồm các kí tự  $'a' \rightarrow 'p'$
- Subtask 3 (25%): Tất cả các từ gồm các kí tự  $'a' \rightarrow 'p'$  và  $'A' \rightarrow 'P'$
- Subtask 4 (25%): Tất cả các từ gồm các kí tự  $'a' \rightarrow 'p'$  và  $'A' \rightarrow 'P'$  và  $'0' \rightarrow '9'$

## Ví dụ

CUBIK.INP	CUBIK.OUT
1 radar	1
1 robot	2
2 FLOW WOLF	2
2 baobab bob	4097

## Lưu ý

- Trong ví dụ đầu tiên, chỉ có một khả năng duy nhất cho từ "radar" xuất hiện trên mỗi cạnh của khối hộp.
- Trong ví dụ thứ hai, có hai khối hộp, chỉ là các phép xoay của nhau - từ "robot" xuất hiện trên mỗi cạnh, và sự khác biệt giữa hai khối hộp là góc dưới trái phía trước chứa  $'r'$  hoặc  $'t'$ .
- Trong ví dụ thứ ba tương tự như ví dụ thứ hai. Việc chúng ta có thể đọc từ trên mỗi cạnh cả hai hướng không ảnh hưởng đến câu trả lời.
- Trong ví dụ thứ tư, chỉ có một cubik kích thước  $1 \times 1 \times 1$  với từ "bob" xuất hiện trên mỗi cạnh và có  $2^{12} = 4096$  khối hộp khác có từ "baobab" xuất hiện trên mỗi cạnh. (Đối với mỗi cạnh trọng số 12 cạnh, chúng ta có hai hướng có thể xuất hiện từ "baobab").

## Bài 9. Sắc màu

File dữ liệu vào:	<code>colors.inp</code>
File kết quả:	<code>colors.out</code>
Hạn chế thời gian:	3 seconds
Hạn chế bộ nhớ:	256 megabytes

Nhắc đến sắc màu, con người ta thường nghĩ tới thế giới tự nhiên với nhiều màu sắc của cây cối, hoa cỏ. Đó là những sắc màu tự nhiên mang lại vẻ đẹp cho cuộc sống. Tại một thành phố nọ, có  $N$  khu vực trồng hoa. Khu vực thứ  $i$ , người dân sẽ trồng duy nhất 1 loài hoa có màu sắc là  $a_i$ , được mô tả bởi 1 số nguyên. Các khu vực được kết nối bởi  $M$  con đường sao cho từ một khu vực bất kỳ có thể di chuyển đến một khu vực khác thông qua các con đường có sẵn.

Để đa dạng hóa vẻ đẹp sắc màu của thành phố, người dân muốn thay thế trồng loại hoa khác ở một số khu vực. Cụ thể, họ mong muốn màu sắc loài hoa trồng ở khu vực thứ  $i$  có màu là  $b_i$ . Việc thay thế trồng loại hoa khác cũng bị hạn chế như sau: Xét 2 khu vực  $u, v$  liên kề (cách nhau bởi 1 con đường), loài hoa ở khu vực  $u$  có thể thay thế bằng loài hoa ở khu vực  $v$  nếu  $a_u > a_v$ . Hãy trả lời xem liệu người dân có thể thay đổi các loài hoa ở một số khu vực như mong muốn được không.

### Dữ liệu vào

- Dòng đầu tiên gồm số nguyên dương  $T$  là số lượng test.
- Với mỗi test:
  - Dòng đầu tiên chứa 2 số nguyên dương  $N, M$  ( $1 \leq N \leq 150000, 0 \leq M \leq 200000$ ). Tổng  $N$  của tất cả các test không vượt quá 300000, tổng  $M$  của tất cả các test không vượt quá 400000.
  - Dòng thứ hai chứa  $N$  số nguyên  $a_1, a_2, \dots, a_N$  ( $1 \leq a_i \leq N$ ).
  - Dòng thứ ba chứa  $N$  số nguyên  $b_1, b_2, \dots, b_N$  ( $1 \leq b_i \leq N$ ).
  - $M$  dòng tiếp theo, mỗi dòng gồm 2 số nguyên  $u, v$  ( $1 \leq u \neq v \leq N$ ) – mô tả một con đường nối 2 khu vực  $u, v$ .

### Kết quả

- Ứng với mỗi test, in ra trên một dòng giá trị 1 nếu có thể đạt được mong muốn hoặc in ra 0 nếu không thể.

### Hạn chế

- Subtask 1 (15%): Đồ thị có dạng hình sao ( $M = N - 1$  và có 1 đỉnh đều liên kề với các đỉnh còn lại). Tổng  $N^2$  của tất cả các test không quá 5000000.
- Subtask 2 (10%):  $N \leq 50$ , đồ thị là đồ thị đầy đủ (tất cả các cặp đỉnh  $u, v$  đều có cạnh kết nối). Tổng  $N \times M$  của tất cả các test không vượt quá 12000000.
- Subtask 3 (20%): Đồ thị có dạng đường thẳng ( $M = N - 1$ ).
- Subtask 4 (15%): Đồ thị có dạng cây,  $\{a_1, a_2, \dots, a_N\}$  là một hoán vị của  $\{1, 2, \dots, N\}$ .
- Subtask 5 (40%): Không có giới hạn gì thêm.

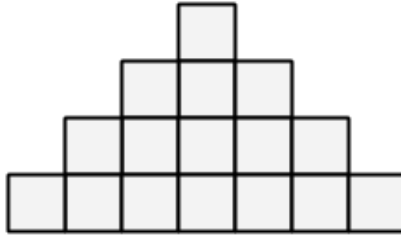


**Ví dụ**

colors.inp	colors.out
2	1
4 4	0
3 3 2 1	
2 1 2 1	
1 2	
2 3	
3 4	
4 2	
4 4	
3 3 2 1	
1 2 2 1	
1 2	
2 3	
3 4	
4 2	

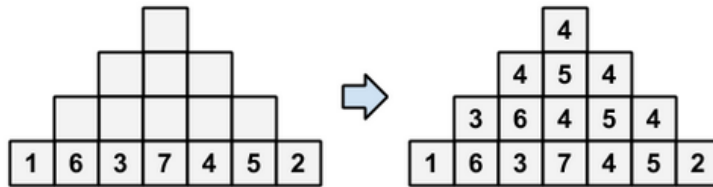
## Bài 10. Kim tự tháp 2 — MEDPYRH

Hùng quan sát một kim tự tháp  $N$  tầng, đánh số từ 1 đến  $N$ , từ đáy lên đỉnh. Tầng thứ  $i$  được ghép bởi  $2 * i - 1$  khối gạch đơn vị hình vuông xếp liên tiếp nhau thành hàng ngang. Kim tự tháp được xây dựng sao cho các khối gạch ở trung tâm của các tầng thẳng hàng với nhau theo chiều dọc.



Hình 1: Một kim tự tháp với  $N = 4$  tầng

Hùng viết  $2 * N - 1$  số nguyên  $A_1, A_2, \dots, A_N$  vào các khối gạch của tầng  $N$ . Sau đó, anh ta viết các số nguyên vào tất cả các khối còn lại, theo quy tắc sau: Số nguyên được viết vào khối gạch  $b$  phải bằng trung vị của ba số nguyên ở ba khối trực tiếp dưới  $b$ , hoặc ở phía dưới ngay bên trái hoặc phía dưới ngay bên phải của  $b$ .



Hình 2: Một kim tự tháp được điền số

Sau đó, anh ta xóa tất cả các số nguyên được viết trong các khối và chỉ nhớ  $2N - 1$  số được ghi ở tầng  $N$ .

**Yêu cầu:** Hãy tìm lại giá trị số nguyên đã được Hùng ghi trên khối gạch tầng 1 của kim tự tháp.

### Dữ liệu vào

- Dòng đầu chứa một số nguyên dương  $T$  ( $T \leq 1000$ ) là số lượng bộ test.
- Mỗi dòng trong số  $T$  nhóm dòng sau gồm:
  - dòng đầu chứa một số nguyên dương  $N$  ( $N \leq 10^5$ );
  - dòng thứ hai chứa  $2N - 1$  số nguyên không âm  $A_1, A_2, \dots, A_N$  ( $A_i \leq 10^9$ ).

### Kết quả

Ghi ra  $T$  dòng, mỗi dòng là kết quả tìm được của bộ test tương ứng trong input.

**Ví dụ**

test	answer
2	4
4	8
1 6 3 7 4 5 2	
5	
1 2 3 7 8 9 6 4 5	

**Hạn chế**

- Trong mọi bộ test, tổng tất cả các giá trị  $N$  không quá  $5 \times 10^5$ ;
- Subtask 1: 30% số điểm có tổng mọi giá trị  $N$  trong mọi bộ test không quá 5000;
- Subtask 2: 30% số điểm khác có  $A_i \in [0, 1]$  với mọi  $i \in [1, 2N - 1]$  trong mọi bộ test;
- Subtask 3: 40% số điểm còn lại không có giới hạn gì thêm.

## Bài 11. HNTOUR - Khám phá Hà Nội

Hà Nội, thủ đô của Việt Nam, không chỉ nổi tiếng với bề dày lịch sử và văn hóa mà còn là điểm đến thu hút du khách với nhiều danh lam thắng cảnh. Những địa điểm nổi bật ở Hà Nội có thể kể đến Hồ Hoàn Kiếm, một hồ nước thơ mộng nằm giữa trung tâm thành phố với Tháp Rùa; đền Ngọc Sơn, nơi gắn liền với những truyền thuyết lịch sử; Văn Miếu - Quốc Tử Giám, trường đại học đầu tiên của Việt Nam, là biểu tượng của tri thức và tinh hoa văn hóa.

Thành phố Hà Nội được chia thành  $n$  khu vực. Có  $n - 1$  tuyến đường hai chiều, tuyến đường thứ  $i$  có độ dài  $w_i$  nối liền hai khu vực  $u_i$  và  $v_i$ . Hệ thống đường đảm bảo tồn tại đường đi giữa bất kì cặp khu vực nào. Nói cách khác, hệ thống tuyến đường của thành phố Hà Nội có dạng đồ thị cây.

Nam, một học sinh của lớp 12A, quyết định dùng kì nghỉ hè của mình để đi du lịch và khám phá mọi ngóc ngách của thành phố Hà Nội. Cậu dự định chọn một khu vực  $s$  để bắt đầu hành trình, sau đó ghé thăm  $n - 1$  khu vực còn lại, mỗi khu vực đúng một lần, rồi quay trở lại khu vực  $s$  để kết thúc chuyến đi.

Khi đang ở khu vực  $u$ , Nam có thể thực hiện một trong hai hành động sau:

- Ghé thăm khu vực  $v$  kề với  $u$  thông qua tuyến đường nối hai khu vực đó. Chi phí của hành động này là  $X * w$  với  $w$  là độ dài của tuyến đường.
- Ghé thăm khu vực  $v$  bất kỳ bằng cách dịch chuyển tức thời đến khu vực đó. Chi phí của hành động này là  $Y$ .

Trong đó,  $X$  và  $Y$  là hai hằng số được cho trước.

Nam muốn biết tổng chi phí tối thiểu để thực hiện chuyến đi khám phá thành phố của mình. Bạn hãy giúp Nam tính toán chi phí này nhé?

### Dữ liệu vào

- Dòng đầu tiên gồm ba số nguyên  $n, X, Y$  ( $2 \leq n \leq 200\,000, 1 \leq X, Y \leq 10^9$ ) cho biết số khu vực của thành phố Hà Nội và hai hằng số  $X, Y$  được đề cập trong đề bài.
- $n - 1$  dòng tiếp theo, dòng thứ  $i$  gồm ba số nguyên  $u_i, v_i$  và  $w_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9$ ) mô tả tuyến đường thứ  $i$ .

Dữ liệu vào đảm bảo tồn tại đường đi giữa bất kì cặp thành phố nào.

### Kết quả

In ra một số nguyên duy nhất là tổng chi phí thấp nhất tìm được

### Ví dụ

HNTOUR

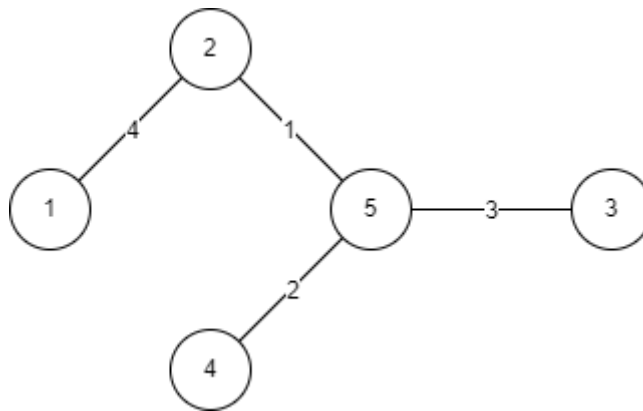
test	answer
5 10 8 1 2 4 2 5 1 5 4 2 5 3 3	40
5 2 8 1 2 4 2 5 1 5 4 2 5 3 3	30

## Hạn chế

- Có 15% số điểm thỏa:  $n \leq 8$
- 15% số điểm tiếp theo thỏa:  $n \leq 16$
- 10% số điểm tiếp theo thỏa:  $u_i = i, v_i = i + 1$  hoặc  $u_i = i + 1, v_i = i$  với mọi  $1 \leq i < n$
- 60% số điểm tiếp còn lại không có giới hạn gì thêm.

## Lưu ý

Hình vẽ minh họa mạng lưới tuyến đường thành phố Hà Nội trong các ví dụ:



Ở ví dụ thứ nhất, Nam có thể xuất phát từ khu vực 1, lần lượt dịch chuyển tức thời đến các khu vực 2, 3, 4, 5 rồi dịch chuyển trở lại khu vực 1. Chi phí cho 5 lần dịch chuyển là  $5 * 8 = 40$ .

Ở ví dụ thứ hai, Nam có thể xuất phát từ khu vực 3 và thực hiện các hành động sau:

- Thăm khu vực 2 bằng cách dịch chuyển tức thời đến đó. Chi phí là 8.
- Thăm khu vực 5 bằng cách đi qua tuyến đường thứ 2. Chi phí là  $2 * 1 = 2$ .
- Thăm khu vực 4 bằng cách đi qua tuyến đường thứ 3. Chi phí là  $2 * 2 = 4$ .
- Thăm khu vực 1 bằng cách dịch chuyển tức thời đến đó. Chi phí là 8.
- Quay trở lại khu vực 3 bằng cách dịch chuyển tức thời đến đó. Chi phí là 8.

Tổng chi phí là:  $8 + 2 + 4 + 8 + 8 = 30$

## Bài 12. Pháo hoa

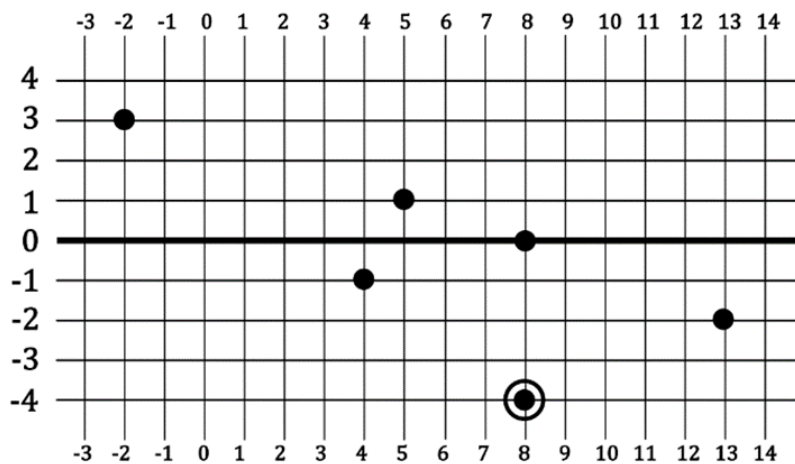
File dữ liệu vào: WATCHFW.INP  
 File kết quả: WATCHFW.OUT  
 Hạn chế thời gian: 1 second  
 Hạn chế bộ nhớ: 256 megabytes

Thành phố T được quy hoạch theo hướng ô bàn cờ: các đường phố được xây dựng như một lưới vuông vô hạn - bất kỳ hai con đường nào cũng song song hoặc vuông góc với nhau và khoảng cách giữa hai đường song song gần nhất là như nhau (ta coi khoảng cách này là 1 đơn vị khoảng cách). Tất cả các đường phố hướng Đông - Tây được gọi là đường ngang và được đánh số bằng các số nguyên liên tiếp theo hướng Nam - Bắc. Tất cả các đường phố hướng Nam - Bắc được gọi là đường dọc và được đánh số bằng các số nguyên liên tiếp theo hướng Tây - Đông. Biết rằng, đường ngang được đánh số 0 được gọi là Đại lộ Trung tâm.

Nhân dịp 60 năm thành lập, thành phố muốn tổ chức một màn bắn pháo hoa tại một ngã tư nơi Đại lộ Trung tâm và một đường dọc  $V$  nào đó. Pháo hoa sẽ được nhìn thấy dọc theo cả hai con đường này. Hơn nữa, vì lý do an toàn, người quan sát phải cách xa ít nhất  $S$  đơn vị so với điểm bắn pháo. Vì vậy, người dân phải đứng xem tại một vị trí trên Đại lộ Trung tâm hoặc đường dọc  $V$  sao cho khoảng cách đến điểm bắn pháo không nhỏ hơn  $S$  đơn vị.

Ví dụ, nếu  $S = 2$ , thì người quan sát có thể đứng ở bất kỳ vị trí nào trên Đại lộ Trung tâm, ngoại trừ những vị trí nằm giữa ngã tư Đại lộ Trung tâm - đường dọc  $V - 1$  và ngã tư Đại lộ Trung tâm - đường dọc  $V + 1$ , và ở bất kỳ vị trí nào trên đường dọc  $V$ , ngoại trừ những vị trí nằm giữa ngã tư đường ngang  $(-1)$  - đường dọc  $V$  và ngã tư đường ngang  $1$  - đường dọc  $V$ .

UBND thành phố biết được vị trí những người dân quan tâm đến sự kiện (họ đều đứng tại các ngã tư), vì vậy, họ muốn tổng khoảng cách từ vị trí của mỗi người đến vị trí gần nhất mà họ có thể quan sát pháo hoa là nhỏ nhất.



Ví dụ, nếu  $S = 2$  và có bảy người quan tâm đang được hiển thị trên bản đồ (có hai người ở vị trí  $(-4, 8)$ ), thì nên chọn bắn pháo hoa tại ngã tư Đại lộ Trung tâm - đường dọc 8. Trong trường hợp này, tổng khoảng cách là 9 đơn vị.

### Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương  $N, S$  ( $1 \leq N \leq 10^5; 1 \leq S \leq 10^6$ );
- $N$  dòng tiếp theo, mỗi dòng chứa hai số nguyên  $h_i, v_i$  ( $|h_i|, |v_i| \leq 10^9$ ) - là vị trí của  $N$  người quan tâm đến sự kiện, người này đang đứng tại ngã tư đường ngang  $h_i$  - đường dọc  $v_i$ .

## Kết quả

- In ra kết quả bài toán.

## Hạn chế

- Subtask 1 (40%):  $S = 1$ .
- Subtask 2 (30%):  $N \leq 5000$ .
- Subtask 3 (30%): Không có giới hạn gì thêm.

## Ví dụ

WATCHFW.INP	WATCHFW.OUT
7 2 3 -2 0 8 -4 8 -1 4 -2 13 -4 8 1 5	9

## Bài 13. BITPUZZLE - Câu đố

Vào hôm giảng dạy về các phép toán xử lý bit trong tin học, thầy dạy tin của lớp 12A đã đưa ra bài toán như sau, đòi hỏi sự kết hợp giữa kiến thức toán học và kiến thức về các phép toán xử lý bit trong tin học để giải được.

Thầy chủ nhiệm viết ra hai dãy số nguyên không âm  $a_1, a_2, \dots, a_n$  và  $b_1, b_2, \dots, b_n$ , mỗi dãy gồm  $n$  phần tử. Sau đó, với mỗi cặp chỉ số  $(i, j)$  sao cho  $1 \leq i, j \leq n$ , thầy viết lên bảng giá trị  $a_i + b_j$  (tổng giữa hai số nguyên). Có thể nhận thấy rằng sẽ có  $n^2$  số nguyên được viết lên bảng. Cuối cùng, thầy tính giá trị  $S$  bằng tổng XOR của tất cả các số nguyên được viết ra. Thầy đố cả lớp viết được chương trình tính nhanh giá trị  $S$  nói trên.

Ở đây, phép XOR (hay "exclusive or") giữa hai số nguyên không âm  $x$  và  $y$ , được ký hiệu là  $x \oplus y$ , là một phép toán nhị phân so sánh từng cặp bit của  $x$  và  $y$ . Với mỗi vị trí bit, nếu hai bit khác nhau, kết quả là 1, còn nếu hai bit giống nhau, kết quả là 0. Ví dụ:  $5 \oplus 3 = 6$  vì trong hệ nhị phân,  $5 = 101_2$  và  $3 = 011_2$ , nên  $101_2 \oplus 011_2 = 110_2 = 6$ .

Các bạn học sinh lớp 12A dù đã rất nỗ lực, nhưng vẫn không thể giải được bài toán hóc búa của thầy dạy tin. Hãy giúp các bạn học sinh lớp 12A viết chương trình để giải bài toán này nhé.

### Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $1 \leq n \leq 100\,000$ ) là độ dài của hai dãy  $a$  và  $b$ .
- Dòng thứ hai gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 2^{20}$ ) cho biết các phần tử của dãy  $a$ .
- Dòng thứ ba gồm  $n$  số nguyên  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i < 2^{20}$ ) cho biết các phần tử của dãy  $b$ .

### Kết quả

In ra một số nguyên duy nhất là giá trị  $S$  cần tìm

### Ví dụ

BITPUZZLE

test	answer
2 1 3 2 4	4
5 3 9 8 7 7 5 10 2 6 4	13
1 100 100	200

### Hạn chế

- Có 20% số điểm thỏa:  $n \leq 1\,000$
- 30% số điểm tiếp theo thỏa:  $a_i, b_i < 2^{10}$
- 50% số điểm còn lại không có giới hạn gì thêm



**Lưu ý**

Ở ví dụ thứ nhất, thầy dạy tin sẽ viết các số sau lên bảng:

- Với  $(i, j) = (1, 1)$ :  $a_1 + b_1 = 1 + 2 = 3$
- Với  $(i, j) = (1, 2)$ :  $a_1 + b_2 = 1 + 4 = 5$
- Với  $(i, j) = (2, 1)$ :  $a_2 + b_1 = 3 + 2 = 5$
- Với  $(i, j) = (2, 2)$ :  $a_2 + b_2 = 3 + 4 = 7$

Tổng XOR của các số trên là:

$$\begin{aligned} S &= 3 \oplus 5 \oplus 5 \oplus 7 \\ &= 011_2 \oplus 101_2 \oplus 101_2 \oplus 111_2 \\ &= 110_2 \oplus 101_2 \oplus 111_2 \\ &= 011_2 \oplus 111_2 \\ &= 100_2 = 4 \end{aligned}$$

## Bài 14. Nhiên liệu

File dữ liệu vào: FFUEL.INP  
 File kết quả: FFUEL.OUT  
 Hạn chế thời gian: 1 second  
 Hạn chế bộ nhớ: 256 megabytes

Trong thời kì giá xăng dầu tăng mạnh, tổ chức O.W.C.A đang cần tính toán số tiền mua nhiên liệu để Perry the Platypus (Agent P) có thể di chuyển từ nơi nhận nhiệm vụ đến nơi Doofenshmirtz đang thực hiện âm mưu xấu xa của mình, cách vị trí nhận nhiệm vụ  $D$  km (Doofenshmirtz Evil Inc.).

Phi cơ của Agent P **bắt đầu** bay với  $F$  lít nhiên liệu, nó tiêu thụ 1 lít/km và có thể chứa bất cứ lượng nhiên liệu nào.

Bên cạnh đó, có tất cả  $N$  trạm nhiên liệu giữa nơi nhận nhiệm vụ đến đích, với trạm thứ  $i$  cách nơi nhận nhiệm vụ  $X_i$  km. Ở mỗi trạm nhiên liệu, Agent P chỉ có thể nạp thêm  $A_i$  lít nhiên liệu, và chỉ được nạp khi  $F \leq B_i$  ( $F$  là lượng nhiên liệu khi **bắt đầu**).

Tổ chức O.W.C.A cần giá trị  $F$  nhỏ nhất có thể sao cho Agent P vẫn có thể đến nơi Doofenshmirtz đang chuẩn bị sử dụng một cỗ máy xấu xa nào đó.

### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên  $N, D$  ( $1 \leq N \leq 3 \times 10^5; 1 \leq D \leq 10^9$ );
- $N$  dòng tiếp theo, mỗi dòng chứa ba số nguyên  $X_i, A_i, B_i$  ( $1 \leq A_i, B_i \leq 10^9; 0 < X_i < D$ ).

### Kết quả

- In ra kết quả bài toán là giá trị  $F$  nhỏ nhất.

### Hạn chế

- Subtask 1 (10%):  $N = 1$ .
- Subtask 2 (15%):  $B_i = 10^9$ .
- Subtask 3 (15%):  $1 \leq D \leq 10^4; 1 \leq N \leq 10^4$ .
- Subtask 4 (15%):  $1 \leq D \leq 10^4$ .
- Subtask 5 (10%):  $1 \leq N \leq 16$ .
- Subtask 6 (15%):  $1 \leq N \leq 10^4$ .
- Subtask 7 (20%): Không có giới hạn gì thêm.

### Ví dụ

FFUEL.INP	FFUEL.OUT
1 10 4 8 6	4
5 100 50 30 25 50 40 25 25 25 25 75 20 25 5 5 25	20

## Lưu ý

Ở ví dụ thứ nhất, phi cơ với  $F = 4$  lít nhiên liệu ban đầu.

- Bay đến trạm nhiên liệu thứ nhất với 4 lít nhiên liệu;
- Nạp thêm 8 lít nhiên liệu (vì  $F \leq B_1 = 6$ );
- Bay đến vị trí của Doof chỉ cần sử dụng 6 lít nhiên liệu.

## Bài 15. SUMMERCAMP - Trại hè

Hoạt động trại hè thường niên đang được tổ chức tại một trường trung học phổ thông nọ. Trong trại hè này, có  $n$  trò chơi được đánh số từ 1 đến  $n$ . Trò chơi thứ  $i$  ( $1 \leq i \leq n$ ) yêu cầu mất  $c_i$  thể lực, và nếu Nam chơi trò chơi đó, độ vui vẻ của cậu ấy sẽ tăng thêm  $h_i$ . Mỗi trò chơi chỉ có thể được tổ chức trong  $p$  đơn vị thời gian vì yêu cầu về thiết bị và không gian. Khi trò chơi thứ  $i$  bắt đầu tại thời điểm  $t_i$ , người tham gia chỉ có thể chơi trò chơi đó từ thời điểm  $t_i$  đến thời điểm  $t_i + (p - 1)$  (bao gồm cả hai thời điểm này). Ngoài ra, mỗi trò chơi chỉ có thể được chơi tối đa một lần.

Nam, học sinh lớp 12A, muốn tham gia trại hè này và chơi một số trò chơi để tối đa hóa độ vui vẻ của mình. Tuy nhiên, vì cậu ấy rất bận rộn, cậu chỉ có thể trại hè một lần, và tại một thời điểm duy nhất. Nói cách khác, nếu Nam lựa chọn tham gia vào thời điểm  $t$ , cậu chỉ có thể chơi các trò chơi đang diễn ra tại thời điểm  $t$ . Tuy nhiên, Nam có thể chơi nhiều trò chơi, miễn là không vượt quá giới hạn thể lực của mình. Cậu không thể chơi cùng một trò chơi nhiều lần theo quy định của trại hè. Lưu ý rằng không cần thiết phải sử dụng hết giới hạn thể lực.

Nam đã lập ra một danh sách gồm  $q$  câu hỏi, câu hỏi thứ  $j$  được biểu diễn bằng cặp số nguyên  $(a_j, b_j)$ , yêu cầu: giả sử Nam lựa chọn tham gia trại hè vào thời điểm  $a_j$ , và có thể tiêu tốn tối đa  $b_j$  thể lực để chơi các trò chơi, hỏi độ vui vẻ tối đa có thể đạt được là bao nhiêu. Hãy giúp Nam trả lời các câu hỏi trên nhé.

### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $p$  ( $1 \leq n \leq 4\,000$ ,  $1 \leq p \leq 10^6$ ) — số lượng trò chơi và thời gian tổ chức mỗi trò chơi.
- $n$  dòng tiếp theo mô tả các trò chơi. Trò chơi thứ  $i$  ( $1 \leq i \leq n$ ) được mô tả bằng ba số nguyên  $c_i$ ,  $h_i$ ,  $t_i$  ( $1 \leq c_i, h_i \leq 4\,000$ ,  $1 \leq t_i \leq 10^6$ ) — thể lực yêu cầu của trò chơi, độ vui vẻ tăng thêm khi tham gia trò chơi, và thời điểm trò chơi bắt đầu.
- Dòng tiếp theo chứa số nguyên  $q$  ( $1 \leq q \leq 20\,000$ ) — số lượng câu hỏi của Nam.
- $q$  dòng tiếp theo mô tả các lượt tham gia. Lượt thứ  $j$  ( $1 \leq j \leq q$ ) được mô tả bằng hai số nguyên  $a_j$ ,  $b_j$  ( $1 \leq a_j \leq 2 \times 10^6$ ,  $1 \leq b_j \leq 4\,000$ ) — thời điểm đến tham gia và giới hạn thể lực cho lượt chơi thứ  $j$ .

### Kết quả

Với mỗi câu hỏi, in ra một số nguyên trên một dòng, là độ vui vẻ tối đa có thể đạt được đối với câu hỏi đó.

### Ví dụ

SUMMERCAMP

test	answer
4 5	0
2 5 2	6
3 6 1	11
4 7 2	12
10 14 6	19
6	0
1 2	
1 5	
2 5	
3 6	
6 13	
100 100	

## Hạn chế

- Có 15% số điểm thỏa:  $n, q \leq 10$
- 15% số điểm tiếp theo thỏa:  $n, q \leq 100$
- 50% số điểm tiếp theo thỏa:  $p, t_i \leq 2\,000; a_j \leq 4\,000$
- 20% số điểm còn lại không có giới hạn gì thêm.

## Lưu ý

Bảng bên dưới mô tả các trò chơi được tổ chức trong ví dụ thứ nhất:

Trò chơi	Thể lực	Độ vui vẻ	Thời điểm										
			1	2	3	4	5	6	7	8	9	10	11
1	2	5		×	×	×	×	×					
2	3	6	×	×	×	×	×						
3	4	7		×	×	×	×	×					
4	10	14						×	×	×	×	×	

- Với câu hỏi đầu tiên, tại thời điểm 1, chỉ có trò chơi 2 được tổ chức. Tổng thể lực của Nam là 2, không đủ để tham gia trò chơi này, nên độ vui vẻ lớn nhất có thể đạt được là 0.
- Với câu hỏi thứ hai, tại thời điểm 1, chỉ có trò chơi 2 được tổ chức. Tổng thể lực của Nam là 5, đủ để tham gia trò chơi này, nên độ vui vẻ lớn nhất có thể đạt được là 6.
- Với câu hỏi thứ ba, tại thời điểm 2, có ba trò chơi 1, 2, 3 được tổ chức. Tổng thể lực của Nam là 5, do đó cậu có thể tham gia hai trò chơi 1 và 2 (yêu cầu tổng thể lực  $2 + 3 = 5$ ) để đạt độ vui vẻ lớn nhất có thể là  $5 + 6 = 11$ .
- Với câu hỏi thứ tư, tại thời điểm 3, có ba trò chơi 1, 2, 3 được tổ chức. Tổng thể lực của Nam là 6, do đó cậu có thể tham gia hai trò chơi 1 và 3 (yêu cầu tổng thể lực  $2 + 4 = 6$ ) để đạt độ vui vẻ lớn nhất có thể là  $5 + 7 = 12$ .
- Với câu hỏi thứ năm, tại thời điểm 6, có ba trò chơi 1, 3, 4 được tổ chức. Tổng thể lực của Nam là 13, do đó cậu có thể tham gia hai trò chơi 1 và 4 (yêu cầu tổng thể lực  $2 + 10 = 12$ ) để đạt độ vui vẻ lớn nhất có thể là  $5 + 14 = 19$ .
- Với câu hỏi thứ sáu, tại thời điểm 100, không có trò chơi nào được tổ chức, nên độ vui vẻ lớn nhất có thể đạt được là 0.

## Bài 16. Trung tâm hội nghị

File dữ liệu vào:	TRAINCENTRE.INP
File kết quả:	TRAINCENTRE.OUT
Hạn chế thời gian:	2 seconds
Hạn chế bộ nhớ:	1024 megabytes

Một đất nước nọ có  $N$  thành phố. Các thành phố được kết nối với nhau bởi  $M$  tuyến tàu cao tốc (cùng chiều dài). Có tất cả  $k$  cán bộ văn phòng chính phủ đang làm nhiệm vụ, người thứ  $i$  đang công tác tại thành phố  $B_i$ , nhưng nhà người đó lại ở thành phố  $A_i$  (mỗi ngày họ sẽ đều sử dụng các tuyến cao tốc để đi làm).

Tất nhiên, cả  $k$  cán bộ đều chọn những hành trình sao cho thời gian di chuyển từ nhà đến nơi làm việc là ngắn nhất. Nếu có nhiều hành trình có cùng thời gian di chuyển, họ sẽ chọn ngẫu nhiên một hành trình. Thủ tướng không hề biết mỗi cán bộ chọn hành trình nào, nhưng ông biết xác suất để cán bộ thứ  $i$  chọn một hành trình đi qua thành phố  $w$  là:

$$E_i(w) = \frac{\text{Số lượng hành trình ngắn nhất từ } A_i \text{ đến } B_i, \text{ đi qua thành phố } w}{\text{Số lượng hành trình ngắn nhất từ } A_i \text{ đến } B_i}$$

Thủ tướng muốn xây dựng một Trung tâm Hội nghị Chính phủ tại ga tàu một thành phố  $w$  sao cho nhiều cán bộ có thể thăm trung tâm này nhất trong hành trình đến vị trí làm việc của họ. Cụ thể, Thủ tướng muốn xây dựng tại thành phố  $w$  sao cho  $\sum_{i=1}^k E_i(w)$  là lớn nhất.

Hãy giúp ông ấy tìm được một thành phố  $w$  thỏa mãn, nếu có nhiều thành phố như vậy, chọn một thành phố bất kì.

Dựa theo địa lí của đất nước, trong quá trình xây dựng các tuyến tàu cao tốc, quy hoạch đảm bảo số lượng hành trình ngắn nhất giữa hai thành phố bất kì không vượt quá  $2^{15}$ .

### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên  $N, M$  ( $1 \leq N \leq 5000; 1 \leq M \leq 40000$ )
- $M$  dòng tiếp theo, mỗi dòng chứa hai số nguyên  $u, v$  ( $0 \leq u, v < N; u \neq v$ )
- tồn tại một tuyến tàu cao tốc nối thành phố  $u$  và thành phố  $v$
- Dòng tiếp theo chứa số nguyên  $k$  ( $1 \leq k \leq 2000$ )
- $k$  dòng tiếp theo, mỗi dòng chứa hai số nguyên  $A_i, B_i$  ( $1 \leq A_i, B_i \leq N$ ).

### Kết quả

- In ra kết quả bài toán

### Hạn chế

- Subtask 1 (15%):  $N \leq 1000; M = N - 1; k = 1$ ; Các tuyến tàu cao tốc và thành phố tạo thành đồ thị dạng đường thẳng.
- Subtask 2 (15%):  $N \leq 1000; M = N - 1; k = 1$ .
- Subtask 3 (20%):  $N \leq 1000; M = N - 1; k \leq 200$ ; Các tuyến tàu cao tốc và thành phố tạo thành đồ thị dạng đường thẳng.
- Subtask 4 (20%):  $N \leq 1000; M = N - 1; k \leq 200$ .

- Subtask 5 (10%):  $N \leq 1000$ ;  $M \leq 8000$ ;  $k \leq 20$ .
- Subtask 6 (20%): Không có giới hạn gì thêm.

**Ví dụ**

TRAINCENTRE.INP	TRAINCENTRE.OUT
5 5 0 1 1 2 2 3 3 4 4 0 2 1 3 2 4	2
5 4 0 1 1 2 2 3 3 4 3 0 2 1 3 2 4	2
6 5 0 2 1 2 2 3 3 4 3 5 2 0 5 1 4	2

## Bài 17. BALLOON

Có  $n$  quả bóng bay, mỗi quả có một trong số 4 màu  $\{R, B, G, Y\}$  xếp thành một cột. Ở mỗi bước được phép chọn thủng dây bóng cùng màu liên tiếp nhau từ 2 quả trở lên. Nếu dây bị chọn thủng có  $k$  quả thì điểm số nhận được ở bước đi đó là  $k^2$ . Các quả bóng ở trên sẽ rơi xuống lấp chỗ trống trong cột sao cho thứ tự trên dưới ban đầu của các quả bóng không bị thay đổi. Ví dụ, cột ban đầu có 10 bóng, tính từ trên xuống cột có cấu hình  $BBGGBBYYYY$ . Để chọn thủng hết dây bóng, ta có thể chọn các quả liên tiếp  $BB, GG, BB, YYY$ . Khi đó, điểm số nhận được sẽ là  $2^2 + 2^2 + 2^2 + 3^2 = 21$ . Tuy nhiên, ta có cách khác để nhận được tổng điểm số cao hơn là  $GG, BBBB, YYY$ , với tổng điểm là  $2^2 + 4^2 + 3^2 = 29$ .

**Yêu cầu:** Cho xâu  $S$  độ dài  $n$  chỉ chứa các ký tự trong tập  $\{R, B, G, Y\}$  biểu diễn màu các quả bóng tính từ trên xuống dưới. Hãy xác định tổng điểm số lớn nhất nhận được khi phá hết bóng trong cột. Nếu không thể phá hết bóng thì tổng điểm là 0.

### Dữ liệu vào

Dòng đầu tiên ghi một số nguyên dương  $T \leq 5$  là số lượng test.

Mỗi dòng trong số  $T$  dòng tiếp theo chứa một xâu  $S$ .

### Kết quả

Mỗi dòng ghi ra một số nguyên là kết quả tìm được tương ứng với test trong dữ liệu vào.

### Ví dụ

test	answer
4	29
BBGGBBYYYY	13
BGGGB	24
BGGBGGGG	0
GBGB	

### Hạn chế

- **Subtask 1 (28 điểm)**  $n \leq 15$ ;
- **Subtask 2 (16 điểm)**  $n \leq 200$ , chỉ có 2 màu  $G, B$  và độ dài các đoạn màu giống nhau liên tiếp là như nhau;
- **Subtask 3 (56 điểm)**  $n \leq 200$ .



## Bài 18. CODING

File dữ liệu vào: **stdin**  
 File kết quả: **stdout**  
 Hạn chế thời gian: 1 giây  
 Hạn chế bộ nhớ: 512 MB

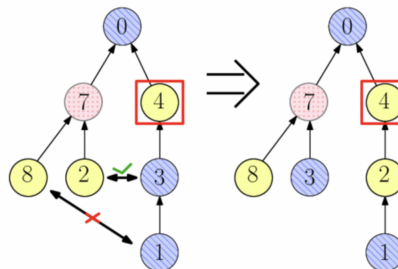
Cho một cây có gốc (gốc tại 0) với  $N$  nút. Mỗi nút biểu thị cho một lập trình viên và được gán một ngôn ngữ lập trình (được đánh số từ 1 đến  $K$ ).  $l_i$  biểu thị ngôn ngữ lập trình được gán cho nút  $i$ .

Để hoàn thành một dự án lập trình, bạn cần chọn một nút  $u$  (trưởng nhóm), sau đó bạn có thể áp dụng thao tác sau không hạn chế số lần:

- chọn hai nút  $v$  và  $w$  có cùng độ sâu trong cây
- hoán đổi ngôn ngữ lập trình được gán cho các nút  $v$  và  $w$

Giá trị của dự án là số nút trong cây con gốc  $u$  có cùng ngôn ngữ lập trình với  $u$ .

**Yêu cầu:** Tìm giá trị lớn nhất của một dự án có thể đạt được và số lượng thao tác tối thiểu cần thiết để đạt được giá trị đó.



### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên,  $N$  và  $K$  là số lượng lập trình viên và số lượng ngôn ngữ lập trình mà nhân viên có thể sử dụng.
- Các nút được đánh số từ 0 đến  $N - 1$  và nút gốc có số 0.
- Dòng tiếp theo chứa  $N$  số nguyên  $l_i$  với  $0 \leq l_i < K$  là ngôn ngữ lập trình được gán cho lập trình viên tương ứng.
- $N - 1$  dòng tiếp theo mô tả cấu trúc cây. Dòng thứ  $i$  chứa số nguyên  $b_i$  với  $0 \leq b_i < N$ , là nút cha trực tiếp của nút  $i$ , với  $i$  trong khoảng từ 1 đến  $N$ .

### Kết quả

Ghi một dòng duy nhất với hai số nguyên,  $P$  và  $S$ , số lượng lập trình viên tối đa (bao gồm cả trưởng nhóm) làm việc trên dự án mới mà bạn có thể đạt được với bất kỳ số lần chuyển đổi nào và số lần chuyển đổi *tối thiểu* cần thiết để đạt được điều này.

**Ví dụ**

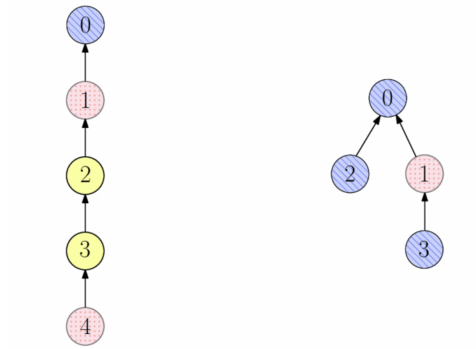
stdin	stdout
5 3 0 1 2 2 1 0 1 2 3	2 0
4 2 0 1 0 0 0 0 1	3 0
9 3 0 0 2 1 2 0 2 1 2 4 8 1 0 4 1 0 7	4 2
8 3 0 2 1 2 2 1 1 1 6 3 0 6 3 0 3	3 2

**Hạn chế**

- $1 \leq N \leq 10^5$ .
- $1 \leq K \leq N$ .
- Subtask 1: Cha trực tiếp của nút  $i$  là  $i - 1$  với mọi  $1 \leq i < N$
- Subtask 2:  $K \leq 2$
- Subtask 3: Đối với mỗi ngôn ngữ lập trình, có nhiều nhất 10 lập trình viên được gán cho ngôn ngữ đó
- Subtask 4:  $N \leq 2000$
- Subtask 5: Không có ràng buộc nào thêm

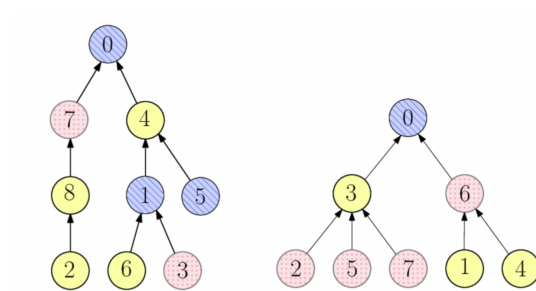
**Giải thích**

Trong hai ví dụ đầu tiên, cấu trúc cây trông như sau, trong đó mẫu mã hóa ngôn ngữ lập trình (0 = “sọc”, 1 = “chấm”, 2 = “đều”):



Trong ví dụ 1, ta có thể chọn lập trình viên 1 làm trưởng nhóm với lập trình viên 4 được gán cùng ngôn ngữ lập trình và không có cách chuyển đổi nào có thể cải thiện điều này.

Trong ví dụ 2, toàn bộ công ty có 3 lập trình viên được gán ngôn ngữ 0, cũng là ngôn ngữ ưa thích của Alice, vì vậy việc chọn Alice làm trưởng nhóm sẽ tạo ra một nhóm có quy mô 3 mà không cần chuyển đổi.



Trong ví dụ 3, ta chọn lập trình viên 4 làm trưởng nhóm và sau đó ta có thể có lập trình viên 1&8 và 2&3

## Bài 19. SPIDERMAN

File dữ liệu vào:	standard input
File kết quả:	standard output
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	256 megabytes

Sau khi có được năng lực Spiderman, Peter Griffin muốn thử sử dụng sức mạnh mới của mình. Thành phố mà cậu đang training có  $n \times m$  tòa nhà, được sắp xếp theo hình bảng. Tòa ở vị trí  $(i, j)$  có độ cao  $a_{ij}$ . Spiderman của Peter Griffin có một năng lực mà các spiderman trước đó không hề có, đó là cậu có thể điều chỉnh được độ đàn hồi của tơ nhện mà mình phóng ra. Từ một tòa nhà cậu có thể đu sang một tòa nhà kề cạnh nếu như lực đàn hồi  $d$  mà cậu sử dụng lớn hơn hoặc bằng chênh lệch chiều cao giữa hai tòa nhà. Peter đã vạch ra một số tòa nhà để cậu xuất phát và luyện đu tơ, để tiết kiệm sức lực, cậu muốn dùng tơ với ít độ đàn hồi nhất có thể. Với mỗi tòa nhà xuất phát, bạn hãy xác định độ đàn hồi  $d$  bé nhất có thể để Peter có thể đu tới ít nhất  $k$  tòa nhà.

### Dữ liệu vào

- Dòng đầu tiên gồm 3 số  $n, m, k$  ( $n, m \leq 500; k \leq n \times m$ ).
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa  $m$  số nguyên, số thứ  $m$  là độ cao của tòa nhà  $(i, j)$  ( $a_{ij} \leq 10^9$ ).
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa  $m$  số nguyên 0/1, trong đó số thứ  $j$  là 1 cho biết ô  $(i, j)$  là một tòa nhà xuất phát.

### Kết quả

Một dòng duy nhất là tổng số độ đàn hồi bé nhất có thể theo yêu cầu bài toán của những tòa nhà xuất phát.

### Ví dụ

standard input	standard output
3 5 10 20 21 18 99 5 19 22 20 16 17 18 17 40 60 80 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1	24

### Lưu ý

Subtask	Tỉ lệ	Ràng buộc
1	20%	$1 \leq n, m \leq 100$
2	20%	$1 \leq a_{ij} \leq 100$
3	60%	Không ràng buộc gì thêm

## Bài 20. Diệt Nấm

File dữ liệu vào:	<code>standard input</code>
File kết quả:	<code>standard output</code>
Hạn chế thời gian:	4 seconds
Hạn chế bộ nhớ:	256 megabytes

Cậu học trò tinh nghịch Tít trong kỳ nghỉ hè sắp tới được ông bà và cậu dao cho công việc diệt nấm trong khu vườn của nhà mình. Tuy tinh nghịch nhưng Tít rất thông minh, cậu để ý thấy những cây nấm này mọc trên  $n$  bãi cỏ khác nhau, trong đó mỗi bãi cỏ có  $a_i$  cây nấm, và chúng có tính chất là mỗi khi cậu tiêu diệt một bãi cỏ với số lượng nấm là  $x$  thì những bãi cỏ chứa  $y$  cây nấm cũng sẽ bị tiêu diệt theo nếu như tồn tại một số nguyên dương  $z$  sao cho bộ ba  $x, y, z$  đôi một nguyên tố cùng nhau và theo thứ tự bất kỳ thỏa mãn được đẳng thức:  $a^2 + b^2 = c^2$ . Bạn hãy giúp Tít trả lời liệu với khu vườn nhà mình thì Tít cần diệt nấm ở ít nhất bao nhiêu bãi cỏ để toàn bộ nấm trong khu vườn bị tiêu diệt?

### Dữ liệu vào

Dòng đầu gồm số tự nhiên  $n$  không vượt quá  $10^6$  là số lượng bãi cỏ trong khu vườn nhà Tít. Tiếp theo là một dòng gồm có  $n$  số nguyên dương đôi một khác nhau không vượt quá  $10^7$  mô tả số lượng nấm trong mỗi bãi cỏ của khu vườn.

### Kết quả

In ra một số duy nhất là số bãi cỏ ít nhất Tít phải tiêu diệt để có thể tiêu diệt toàn bộ nấm trong khu vườn.

### Ví dụ

standard input	standard output
1 2	1
9 7 10 6 5 4 2 9 1 3	7
7 4 2 6 10 8 5 1	6

### Lưu ý

This is the unordered list:

- **Subtask 1**  $n \leq 1000$
- **Subtask 2** Không có giới hạn gì thêm

## Bài 21. Du lịch Hạ Long

File dữ liệu vào:	BAICHAYNM.INP
File kết quả:	BAICHAYNM.OUT
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	256 megabytes

Austin là một khách du lịch rất muốn trải nghiệm du lịch ở Việt Nam. Sau thời gian dài cày cuốc kiếm tiền, cậu đã tự thưởng cho bản thân một chuyến đi Quảng Ninh. Sau một chặng bay dài từ Melbourne về đến Nội Bài, rồi tiếp tục từ Nội Bài về Vân Đồn và một đêm nghỉ ngơi tại "LivinHomestay Ha Long", cậu đã quyết định hôm nay sẽ tắm biển tại Bãi Cháy. Cậu cũng biết được một chợ đêm sẽ lên đèn tại đây vào buổi tối. Vì vậy, cậu muốn di chuyển ngay bây giờ để có thể nhanh chóng chiêm ngưỡng vẻ đẹp của bờ biển và đời sống sinh hoạt nơi đây.

Có thể biểu diễn thành phố Hạ Long dưới dạng một mặt phẳng 2 chiều. Austin hiện đang ở homestay tọa độ  $(s_x, s_y)$ , và chợ đêm Bãi Cháy nằm ở tọa độ  $(e_x, e_y)$ . Austin sẽ đi bộ theo các đường song song với trục tọa độ. Ngoài ra, có  $N$  tòa nhà hình chữ nhật có cạnh song song với trục tọa độ, Austin có thể đi trên các cạnh của nó, nhưng không được phép đi qua bất cứ vị trí nào nằm bên trong nó.

Do số lượng tòa nhà lớn, Austin gặp khá nhiều khó khăn với việc tìm một đường di chuyển ngắn nhất. Hãy giúp Austin tìm đường đi ngắn nhất từ homestay đến chợ đêm Bãi Cháy.

Ta có thể đảm bảo rằng các tọa độ  $x$  là phân biệt, và các tọa độ  $y$  phân biệt. Đồng thời không có hình chữ nhật nào giao nhau, và vị trí của homestay và chợ đêm không nằm trong bất kỳ hình chữ nhật nào.

### Dữ liệu vào

- Dòng đầu tiên chứa năm số nguyên dương  $N, s_x, s_y, e_x, e_y$  ( $0 < N \leq 250000; 0 \leq s_x, s_y, e_x, e_y \leq 10^8$ );
- $N$  dòng tiếp theo, dòng thứ  $i$  chứa bốn số nguyên  $a_i, b_i, c_i, d_i$  ( $0 \leq a_i < c_i \leq 10^8; 0 \leq b_i < d_i \leq 10^8$ ) biểu thị tòa nhà chọc trời thứ  $i$  là một hình chữ nhật có tọa độ 4 góc là  $(a_i, b_i)$ ,  $(a_i, d_i)$ ,  $(c_i, b_i)$ ,  $(c_i, d_i)$ .
- $X = \{s_x, e_x, a_1, a_2, \dots, a_N, c_1, c_2, \dots, c_N\}$ , các phần tử trong  $X$  phân biệt.
- $Y = \{s_y, e_y, b_1, b_2, \dots, b_N, d_1, d_2, \dots, d_N\}$ , các phần tử trong  $Y$  phân biệt.

### Kết quả

- Đường đi ngắn nhất từ homestay đến chợ đêm.

### Hạn chế

- Subtask 1 (30%):  $N \leq 500$ .
- Subtask 2 (30%):  $N \leq 5000$ .
- Subtask 3 (40%): Không có giới hạn gì thêm.

### Ví dụ

BAICHAYNM.INP	BAICHAYNM.OUT
3 2 14 5 1 4 6 6 10 0 7 3 9 1 2 8 5	20
1 0 500 100 503 1 0 99 1000	1097
2 2 8 10 3 3 6 6 10 7 1 8 7	15

## Bài 22. GEMSTONE - Chế tác đá quý

File dữ liệu vào: `stdin`  
 File kết quả: `stdout`  
 Hạn chế thời gian: 1 giây  
 Hạn chế bộ nhớ: 512 MB

Một cuộc thi chế tác đá quý có  $N$  viên đá quý trong bộ sưu tập được đánh số từ 1 đến  $N$ . Mỗi viên đá quý được đặc trưng bởi ba số nguyên  $X$ ,  $Y$  và  $Z$  đại diện cho màu sắc, độ trong suốt, và độ sáng của nó.

Từ một viên đá quý thợ giả kim có thể chế tác biến đổi diện mạo thành viên đá khác nếu như một viên đá có ít nhất một thuộc tính có giá trị bằng với ít nhất một thuộc tính của viên đá thứ hai. Ví dụ, viên đá quý  $(2, 4, 5)$  có thể biến thành viên đá quý  $(4, 3, 3)$  vì cả hai đều có một thuộc tính bằng 4.

Dũng tham gia cuộc thi và bốc phải hai viên đá quý đánh số  $A$  và  $B$ . Nhiệm vụ của Dũng là chế tác sao cho viên đá  $A$  có diện mạo giống với viên đá  $B$  dựa vào thông số thuộc tính của các viên đá quý khác trong kho. Dũng phải thực hiện các phép biến đổi liên tiếp bắt đầu từ viên đá  $A$ , mỗi lần thay đổi các thuộc tính của nó để thành một viên đá khác có các thông số giống với một viên đá trong kho.

**Yêu cầu:** hãy giúp Dũng đề ra phương án chế tác sao cho thực hiện ít số lần biến đổi nhất.

### Dữ liệu vào

- Dòng đầu tiên chứa số nguyên  $T$  là số lượng bộ test.
- Đối với mỗi bộ test:
  - Dòng đầu tiên chứa số nguyên  $N$  là số lượng đá quý trong bộ sưu tập.
  - Dòng thứ hai chứa hai số nguyên  $A$  và  $B$  là số thứ tự của hai viên đá cần biến đổi.
  - Mỗi dòng trong số  $N$  dòng tiếp theo chứa ba số nguyên  $X$ ,  $Y$ ,  $Z$  mô tả các thuộc tính của mỗi viên đá.

### Kết quả

Ghi ra  $T$  dòng, mỗi dòng tương ứng với mỗi bộ test chứa số lần biến đổi ít nhất tìm được hoặc  $-1$  nếu quá trình không thể thực hiện được.

### Ví dụ

stdin	stdout
2	2
4	-1
1 2	
2 1 1	
5 3 6	
4 3 5	
3 2 7	
4	
1 3	
2 1 1	
2 2 2	
4 3 5	
2 2 7	

### Hạn chế

- $1 \leq T \leq 5$
- $2 \leq N \leq 100,000$
- $1 \leq A, B \leq N$  và  $A \neq B$
- $1 \leq X, Y, Z \leq 500,000$



- Tổng các giá trị  $N$  trong tất cả các bộ test không vượt quá 300,000.
- Đảm bảo rằng luôn cần ít nhất một lần biến đổi (hai viên đá  $A$  và  $B$  không có diện mạo giống nhau ngay từ đầu).
- **Subtask 1:** 40% số test có  $N \leq 10$
- **Subtask 2:** 20% số test có  $N \leq 1000$
- **Subtask 3:** 40% số test còn lại không có ràng buộc nào thêm.

## Bài 23. ESCAPING - Tẩu thoát

File dữ liệu vào: `stdin`  
 File kết quả: `stdout`  
 Hạn chế thời gian: 1 giây  
 Hạn chế bộ nhớ: 512 MB

$M$  người bạn gồm Huy và bạn của Huy tham quan một khu vườn hình chữ nhật có kích thước  $w \times h$ . Trong vườn có  $N$  chậu cây. Khu vườn có 4 lối ra vào ở 4 góc (1 = góc trái dưới, 2 = góc phải dưới, 3 = góc phải trên, 4 = góc trái trên). Góc trái dưới có tọa độ  $(0,0)$ , góc phải trên có tọa độ  $(w,h)$ . Chậu cây thứ  $i$  được biểu diễn bằng một đường tròn đặt tại tọa độ  $(x_i, y_i)$ , bán kính  $r_i$ . Người bạn thứ  $j$  tham quan từ lối vào thứ  $e_j$ , được biểu diễn bằng một đường tròn bán kính  $R_j$  tiếp xúc với hai cạnh của hình chữ nhật ứng với góc  $e_j$ .

Mọi người có thể tham quan thỏa thích trong khu vườn, nhưng không được va chạm vào những chậu cây. Với mỗi người bạn, hãy tìm xem người đó có thể rời khu vườn bằng những lối ra nào.

### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương  $N, M$  ( $1 \leq N \leq 2000; 1 \leq M \leq 10^5$ );
- Dòng thứ hai chứa hai số nguyên dương  $w, h$  ( $4R_{max} \leq w, h \leq 10^9$ ), với  $R_{max}$  là bán kính lớn nhất trong những người bạn.
- $N$  dòng tiếp theo, mỗi dòng chứa ba số nguyên  $x_i, y_i, r_i$  (chậu cây sẽ không giao với biên của khu vườn);
- $M$  dòng cuối cùng, mỗi dòng chứa hai số nguyên  $R_j, e_j$  ( $1 \leq R_j \leq \min(w, h); 1 \leq e_j \leq 4$ ).

Dữ liệu đảm bảo không có chậu cây nào giao với hình vuông kích thước  $2R_{max} \times 2R_{max}$  ở mỗi góc.

### Kết quả

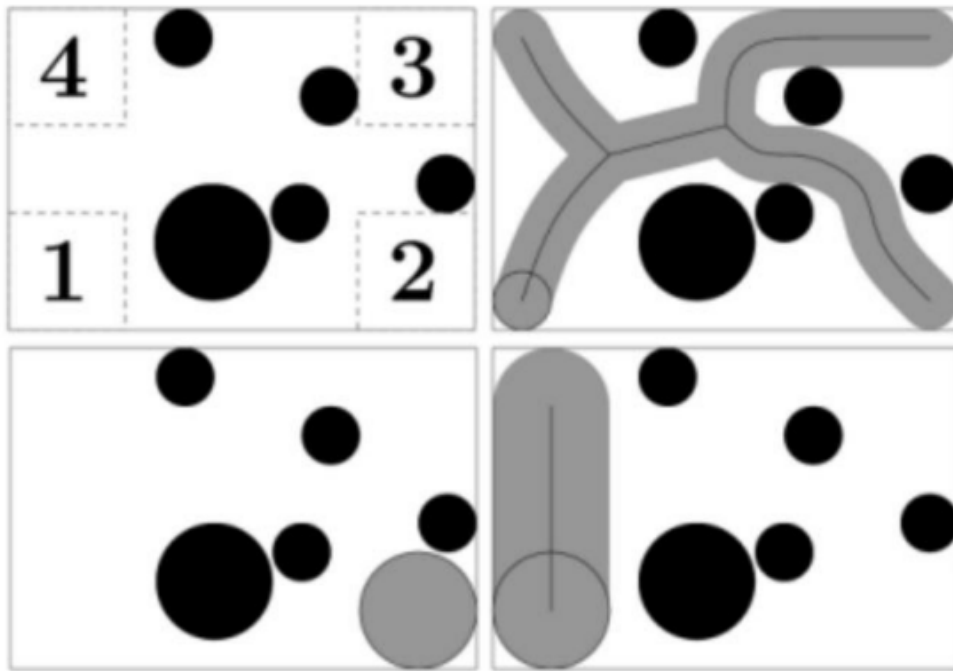
In kết quả trên  $M$  dòng, mỗi dòng chứa một số các số nguyên đã sắp xếp tăng dần, sát nhau là kí hiệu của các lối ra của người thứ  $i$ .

### Ví dụ

stdin	stdout
5 3	1234
16 11	2
11 8 1	14
6 10 1	
7 3 2	
10 4 1	
15 5 1	
1 1	
2 2	
2 1	

### Giải thích

Hình dưới minh họa đường đi và các lối ra vào cho từng người bạn.



## Hạn chế

- **Subtask 1:** 33% số test ứng với 33% số điểm có  $1 \leq N \leq 2000, M = 1$ .
- **Subtask 2:** 33% số test khác ứng với 33% số điểm có  $1 \leq N \leq 200, 1 \leq M \leq 10^5$ .
- **Subtask 3:** 34% test còn lại ứng với 34% số điểm có  $1 \leq N \leq 2000, 1 \leq M \leq 10^5$ .

## Bài 24. LANDSLIDE - Sạt lở đất

File dữ liệu vào:	<code>stdin</code>
File kết quả:	<code>stdout</code>
Hạn chế thời gian:	1 giây
Hạn chế bộ nhớ:	512 MB

Vương quốc Hạnh Phúc có  $N$  ngôi làng (được đánh số từ 1 đến  $N$ ), các ngôi làng được nối với nhau bằng  $N - 1$  con đường nối trực tiếp giữa hai ngôi làng. Các con đường có độ dài khác nhau. Các con đường đảm bảo có đường đi đến nhau giữa hai ngôi làng bất kì. Trong  $N$  ngôi làng, có  $S$  ngôi làng có kho lương thực.

Do ảnh hưởng của cơn bão số 2, một con đường trong số  $N - 1$  con đường có thể bị sạt lở và không thể đi lại được. Để đối phó với tình hình, quốc vương của vương quốc Hạnh Phúc cho đặt một trạm cứu hộ trực thăng ở ngôi làng  $H$ . Người dân khi đi đến làng  $H$  thì sẽ thoát khỏi ảnh hưởng của cơn bão và đến được nơi an toàn.

Để chuẩn bị cho các tình huống khác nhau của cơn bão, quốc vương cần trả lời được  $Q$  câu hỏi khác nhau. Mỗi câu hỏi sẽ có dạng nếu con đường  $I$  bị sạt lở, người dân của làng  $R$  có thể đến được trạm cứu hộ được không? Nếu người dân làng  $R$  không thể đến được trạm cứu hộ thì khoảng cách ngắn nhất từ  $R$  đến một làng có kho lương thực bất kì là bao nhiêu?

### Dữ liệu vào

Dòng đầu tiên của dữ liệu đầu vào chứa 4 số nguyên  $N, S, Q$  và  $H$ , với  $N$  là số lượng ngôi làng,  $S$  ( $1 \leq S \leq N$ ) là số lượng làng có kho lương thực,  $Q$  là số truy vấn cần trả lời và  $H$  là làng có trạm cứu hộ trực thăng.

Mỗi dòng trong  $N - 1$  dòng tiếp theo có chứa ba số nguyên  $U, V$  và  $W$ , có ý nghĩa là tồn tại một con đường độ dài  $W$  ( $1 \leq W \leq 10^9$ ) nối trực tiếp giữa làng  $U$  và làng  $V$  ( $1 \leq U, V \leq N$ ).

Mỗi dòng trong  $S$  dòng tiếp theo chứa một số nguyên duy nhất  $C$ , có ý nghĩa là làng  $C$  ( $1 \leq C \leq N$ ) có kho lương thực. Chú ý dữ liệu đảm bảo các số nguyên ở  $S$  dòng này là đôi một khác nhau.

Mỗi dòng trong  $Q$  dòng tiếp theo chứa hai số nguyên  $I$  và  $R$  ( $1 \leq I < N, 1 \leq R \leq N$ ), có ý nghĩa là xét tình huống con đường thứ  $I$  từ dữ liệu đầu vào bị sạt lở và không thể sử dụng được, người dân từ làng  $R$  có thể đến được trạm cứu hộ trực thăng hay không, nếu không thì khoảng cách ngắn nhất từ làng  $R$  đến một làng có kho lương thực là bao xa.

### Kết quả

Dữ liệu đầu ra gồm  $Q$  dòng. Dòng thứ  $i$  chứa câu trả lời của truy vấn thứ  $i$ . Cụ thể hơn, dòng đó sẽ là xâu `escaped` nếu như người dân từ làng tương ứng có thể đến được trạm cứu hộ trực thăng, nếu không thể, dòng đó sẽ là khoảng cách đến làng có kho lương thực gần nhất, hoặc in ra xâu `oo` nếu không có làng nào có thể đến được.

**Ví dụ**

stdin	stdout
5 2 3 5 5 1 3 5 3 2 3 4 1 3 2 2 1 4 2 2 2 1 4 2	3 escaped oo
stdin	stdout
10 2 5 9 7 1 3 9 2 3 10 5 1 8 7 3 10 1 3 5 6 2 2 1 2 3 1 1 4 2 2 2 7 6 1 7 7 8 9 2 6 1 3	escaped 0 escaped 8 escaped

**Hạn chế**

- **Subtask 1:** 10% số test ứng với 10% số điểm có  $1 \leq N \leq 100$ ,  $1 \leq Q \leq 10000$ , và tồn tại đường đi giữa  $U$  và  $V$  khi và chỉ khi  $|A - B| = 1$ .
- **Subtask 2:** 30% số test khác ứng với 30% số điểm có  $1 \leq N \leq 1000$ ,  $1 \leq Q \leq 1000$ .
- **Subtask 3:** 15% số test khác ứng với 15% số điểm có  $1 \leq N \leq 100000$ ,  $1 \leq Q \leq 100000$  và  $S = N$ .
- **Subtask 4:** 45% số test còn lại ứng với 45% số điểm có  $1 \leq N \leq 100000$ ,  $1 \leq Q \leq 100000$ .