

Russia contest 9

Created: 5/1/2025 21:23

Updated: 5/1/2025 21:25

Exported: 5/1/2025 21:25

Bài A: Tối Ưu Hóa Ma Trận

 **Giới hạn thời gian:** 1 giây

 **Giới hạn bộ nhớ:** 256 MB

 **Tên file vào:** chuẩn (standard input)

 **Tên file ra:** chuẩn (standard output)

Đề bài (dịch tiếng Việt)

Ma trận được điều khiển bởi một trí tuệ nhân tạo vô cùng phức tạp. Vì vậy, "Kiến trúc sư" quyết định tối ưu hệ thống này.

Hệ thống có thể được biểu diễn như một chuỗi gồm n nút tính toán được kết nối liên tiếp. Mỗi nút có một độ quan trọng tính toán là w_i (gọi là **tầm quan trọng tính toán**).

Kiến trúc sư có thể **chọn một nút bất kỳ** và **lan tỏa ảnh hưởng** của nút đó sang **k nút bên trái** và **k nút bên phải**. Nếu ở một phía không có đủ k nút, thì chỉ lan sang được số nút hiện có.

Gọi tổng số nút bị ảnh hưởng là d , khi đó:

- Tầm quan trọng tính toán của **nút được chọn** sẽ trở thành:
 $(1 + d) \cdot w_i$
- Tầm quan trọng tính toán của **tất cả các nút bị ảnh hưởng** sẽ trở thành **0**.

Yêu cầu:

Tính **tổng tầm quan trọng tính toán lớn nhất của toàn hệ thống** sau khi thực hiện thao tác trên đúng **một lần**.



Dữ liệu vào

- Dòng 1: Hai số nguyên n và k — số lượng nút và bán kính ảnh hưởng
 $1 \leq n \leq 2 \cdot 10^5, 0 \leq k \leq 2 \cdot 10^5$
- Dòng 2: n số nguyên w_1, w_2, \dots, w_n — độ quan trọng ban đầu của từng nút
 $1 \leq w_i \leq 5 \cdot 10^5$



Dữ liệu ra

- Một số nguyên duy nhất — **giá trị tâm quan trọng lớn nhất có thể đạt được** sau khi lan truyền ảnh hưởng từ một nút.



Tóm tắt đề bài

- Có mảng w gồm n phần tử.
- Chọn 1 vị trí i , ảnh hưởng đến $\leq k$ phần tử bên trái và $\leq k$ phần tử bên phải của nó (nếu có).
- Với mỗi lựa chọn:
 - Các phần tử bị ảnh hưởng biến thành 0.
 - Phần tử trung tâm w_i trở thành $(1 + d) \cdot w_i$, với d là số lượng phần tử bị ảnh hưởng.
- Tính tổng giá trị của toàn mảng sau thao tác, tìm giá trị **lớn nhất có thể**.

Subtasks (Các phần kiểm tra)

Phần kiểm tra	Điểm	Ràng buộc thêm	Yêu cầu phụ thuộc
1	10	$n \leq 3$	—
2	18	$n \leq 6$	1
3	20	$k = 0$	—
4	28	$k \leq 1000$	2
5	24	Không giới hạn thêm	1 - 4

Ví dụ

Input 1:

```
3 1
1 3 2
```

Output 1:

```
9
```

Giải thích: Chọn nút thứ 2: ảnh hưởng đến 2 nút xung quanh $\rightarrow d = 2, (1 + 2) \cdot 3 = 9$, hai nút còn lại là 0.

Input 2:

```
5 1
1 5 3 2 4
```

Output 2:

```
21
```

Giải thích: Chọn nút ở giữa, tối ưu giá trị hệ thống.

Bạn có muốn mình viết luôn đoạn code để giải bài này không?

Bài B: Cần Nhiều Năng Lượng Hơn

 **Giới hạn thời gian:** 1 giây

 **Giới hạn bộ nhớ:** 1024 MB

 **Tên file vào:** chuẩn (standard input)

 **Tên file ra:** chuẩn (standard output)

Đề bài (dịch tiếng Việt)

Các cỗ máy trong Ma Trận tiêu thụ một lượng năng lượng không tưởng. Một số người, khi tương tác trong Ma Trận, giải phóng rất nhiều năng lượng. Do đó, Kiến trúc sư đang lên kế hoạch xây dựng **một căn phòng mới** để đặt các **buồng đông lạnh** chứa người.

Phòng này có **chiều dài** n và **chiều rộng** x , dọc theo cạnh đáy (ngang) của phòng có một **pin năng lượng lớn**.

Trong mỗi hàng của phòng (từ hàng $i = 1$ đến n), sẽ đặt **một buồng đông lạnh** tại một khoảng cách p_i từ bức tường bên dưới. Khoảng cách này thỏa mãn $1 \leq p_i \leq x$.

Một buồng được gọi là "**hiệu quả về năng lượng**" nếu:

- Buồng bên trái (nếu có) **nằm gần hơn** pin so với buồng hiện tại.
- Buồng bên phải (nếu có) **nằm gần hơn** pin so với buồng hiện tại.

Tức là:

$$p_{i-1} < p_i \text{ và } p_i > p_{i+1}$$

Gọi một **cấu hình** là một dãy p_1, p_2, \dots, p_n thỏa mãn điều kiện $1 \leq p_i \leq x$.

 **Yêu cầu:**

Hãy đếm có **bao nhiêu cấu hình** mà trong đó **chính xác có k** buồng được đặt một cách **hiệu quả về năng lượng** như mô tả trên.

Vì kết quả có thể rất lớn, hãy **in ra theo modulo $10^9 + 7$** .

Dữ liệu vào

- Một dòng duy nhất chứa 3 số nguyên n, x, k
 $1 \leq n, x, k \leq 500$

Dữ liệu ra

- Một số nguyên duy nhất — **số lượng cấu hình thỏa mãn**, chia dư cho $10^9 + 7$
-

Tóm tắt đề bài

- Có n hàng, mỗi hàng đặt một buồng ở độ cao từ 1 đến x .
- Cần đếm số cách chọn dãy $p_1, \dots, p_n \in [1, x]$ sao cho có đúng k vị trí i mà:

$$p_{i-1} < p_i > p_{i+1}$$

- Biên p_0 và p_{n+1} **không tồn tại**, nên không xét hiệu quả tại rìa.
- Trả lời modulo $10^9 + 7$

Subtasks (Các phần kiểm tra)

Phần kiểm tra	Điểm	Giới hạn thêm	Phụ thuộc
1	20	$n, x, k \leq 10$	—
2	35	$n, x, k \leq 50$	1
3	15	$n, x, k \leq 300$	2
4	10	$n, x, k \leq 400$	3
5	20	Không giới hạn thêm	4

Ví dụ

Input 1:

4 2 1

Output 1:

4

Input 2:

10 5 4

Output 2:

351596

Bạn có muốn mình viết luôn hướng tiếp cận hoặc code động cho bài này không?

Bài C: Cuộc Đối Đầu Cuối Cùng

 **Giới hạn thời gian:** 1 giây

 **Giới hạn bộ nhớ:** 256 MB

 **Tên file vào:** chuẩn (standard input)

 **Tên file ra:** chuẩn (standard output)

Đề bài (dịch tiếng Việt)

Neo và Trinity đang chống lại đội quân robot đang chiếm lĩnh thành phố ảo San Francisco. Thành phố này được mô tả như một **đường thẳng vô hạn**. Quân địch **tấn công theo từng đợt**, mỗi đợt là **một đoạn thẳng** được lấp đầy robot.

Có **n đợt tấn công**, mỗi đợt i chiếm một đoạn $[l_i, r_i]$ trên trục tọa độ.

🔍 Trinity cho rằng một **nhóm liên tiếp các đợt** từ đợt a đến đợt b (với $a \leq b$) là **nguy hiểm** nếu:

- Tất cả các đoạn từ a đến b có phần giao nhau,
- Độ dài phần giao đó nằm trong đoạn từ m_1 đến m_2 , tức là:

$$m_1 \leq \text{độ dài giao của } [l_a, r_a] \cap [l_{a+1}, r_{a+1}] \cap \dots \cap [l_b, r_b] \leq m_2$$

Ví dụ:

- $n = 3$, các đoạn: $[4, 7], [5, 8], [3, 6]$
- Giao 2 đoạn đầu là $[5, 7]$ (độ dài = 2) \rightarrow thỏa mãn nếu $m_1 \leq 2 \leq m_2$
- Giao 3 đoạn là $[5, 6]$ (độ dài = 1)

Nhiệm vụ của bạn là đếm **số lượng nhóm liên tiếp các đợt tấn công nguy hiểm**.



Dữ liệu vào

- Dòng 1: ba số nguyên n, m_1, m_2 — số đợt tấn công và giới hạn độ dài phần giao
 $1 \leq n \leq 2 \cdot 10^5, 0 \leq m_1 \leq m_2 \leq 10^9$
- n dòng tiếp theo: mỗi dòng gồm hai số nguyên l_i, r_i — đoạn chiếm đóng của đợt thứ i
 $0 \leq l_i < r_i \leq 10^9$



Dữ liệu ra

- Một số nguyên duy nhất — số lượng **nhóm đợt tấn công liên tiếp** có phần giao nằm trong đoạn $[m_1, m_2]$



Tóm tắt đề bài

- Cho n đoạn $[l_i, r_i]$ trên trục số.
- Với mọi đoạn con liên tiếp từ i đến j , tính phần giao các đoạn từ i đến j
- Nếu độ dài giao $\in [m_1, m_2]$, thì nhóm đó được tính là **nguy hiểm**
- Đếm tổng số nhóm như vậy.

Lưu ý:

- Có thể có tới $\approx 2 \times 10^5$ đoạn.
 - Không thể duyệt toàn bộ $\mathcal{O}(n^2)$, cần dùng **kỹ thuật hai con trỏ (two pointers)** hoặc **deque** để giữ min/max động.
-

Subtasks (Các phần kiểm tra)

Phần kiểm tra	Điểm	Ràng buộc thêm	Phụ thuộc
1	12	$n \leq 100$	—
2	18	$l_i \leq l_{i+1}$	—
3	18	$n \leq 1000$	1
4	20	$r_i - l_i \leq 30$ với mọi i	—
5	32	Không giới hạn thêm	1 - 4

Ví dụ

Input 1:

```
4 1 2
0 4
1 5
2 6
3 4
```

Output 1:

```
5
```

Giải thích:

- Mỗi nhóm chứa đợt 4 đều có độ dài phần giao là 1 → hợp lệ
- Nhóm từ 1 đến 3: giao là $[2, 4]$ → độ dài 2 → hợp lệ
- Tổng: 5 nhóm

Input 2:

```
6 0 1
0 4
0 2
0 1
3 4
2 4
0 4
```

Output 2:

Bạn có muốn mình triển khai luôn ý tưởng và code sử dụng hai con trỏ cho bài này không?

D Bài D: Ma Trận Phân Tán

 **Giới hạn thời gian:** 1 giây

 **Giới hạn bộ nhớ:** 256 MB

 **Tên file vào:** chuẩn (standard input)

 **Tên file ra:** chuẩn (standard output)

Đề bài (dịch tiếng Việt)

Trong tình trạng thiếu hụt năng lượng, hệ thống Ma Trận đã được sửa đổi để tiêu thụ ít năng lượng nhất có thể. Có **n nút** trong Ma Trận, đánh số từ 1 đến n , với:

- Nút số **1** là **máy phát điện**, được kết nối ngay từ **thời điểm 0**.
- Các nút khác **kết nối dần** đến các nút đã có năng lượng, tạo thành **cây**, **không có chu trình**.

Các nút **có thể bị hỏng và khôi phục lại** theo thời gian. Một khi nút đã kết nối, nó **không bao giờ kết nối lại** với nút khác.

Khái niệm **độ không ổn định (độ không tin cậy)** của một nút là **số giây đã trôi qua kể từ lần cuối nó được phục hồi** (hoặc kể từ khi được kết nối lần đầu, nếu chưa từng bị hỏng).

Các loại sự kiện (theo thứ tự thời gian, mỗi giây một sự kiện)

Tổng cộng có m sự kiện, mỗi sự kiện ở dòng riêng:

1. $\text{! } x \ y$

⇒ Kết nối nút y đến nút x .

⚠️ Bảo đảm x đã có năng lượng, y chưa từng kết nối trước đó.

2. $\text{- } x$

⇒ Nút x bị hỏng, ngừng dẫn năng lượng.

⚠️ Chỉ gọi khi nút x đang hoạt động.

3. $\text{+ } x$

⇒ Nút x được khôi phục.

⚠️ Chỉ gọi khi trước đó đã bị hỏng.

4. $\text{? } x \ y$

⇒ Hỏi: Cả hai nút x, y có đang nhận năng lượng không?

- Nếu **không**, in -1
- Nếu **có**, in tổng độ không ổn định của **mọi nút trên đường đi từ gốc (nút 1) đến x và từ gốc đến y** .

Dữ liệu vào

- Dòng đầu: hai số nguyên n, m — số nút và số sự kiện
 $2 \leq n, m \leq 2 \cdot 10^5$
- m dòng tiếp theo: mô tả từng sự kiện theo các định dạng trên



Dữ liệu ra

- Với mỗi truy vấn $? x y$, in một dòng:
 - -1 nếu một trong hai không có điện
 - hoặc tổng độ không ổn định của các nút nằm trên đường dẫn từ gốc đến x hoặc y (lấy **hợp** hai đường)
-



Ý tưởng tóm tắt

- Mô hình là **cây**, thêm dần các cạnh theo thời gian qua sự kiện $!$.
- Mỗi nút lưu:
 - Cha** trong cây
 - Thời điểm được kết nối** hoặc **khôi phục lần cuối**
 - Trạng thái **hoạt động** hay **hỏng**
- Với truy vấn $? x y$:
 - Nếu đường từ **nút 1 đến x** hoặc **nút 1 đến y** có nút **bị hỏng** \rightarrow trả -1
 - Nếu không: hợp hai đường, lấy tất cả các đỉnh, tính thời gian hiện tại – thời khởi động mỗi nút



Có thể dùng **DSU on Tree / HLD / Binary Lifting** để đi nhanh từ x đến gốc.



Subtasks (Các phần kiểm tra)



Phần kiểm tra	Điểm	Ràng buộc thêm	Phụ thuộc
1	12	$n, m \leq 1000$, không có `-' và `+' (không hỏng)	—
2	14	$n, m \leq 1000$	1
3	18	Không có `-' và `+'	1
4	24	Không có `+' (đã hỏng thì không khôi phục lại)	3
5	32	Không giới hạn	1-4

Ví dụ

Input:

```
3 7
! 1 2
! 1 3
? 2 3
- 3
? 2 3
+ 3
? 2 3
```

Output:

```
6
-1
14
```

Giải thích:

- T1: 1 được bật từ đầu (thời 0), 2 được bật tại $t=1$, 3 tại $t=2 \rightarrow$ `? 2 3` tại $t=3 \rightarrow$ tổng = $3 + 2 + 1 = 6$
- T4: tắt 3 \rightarrow `? 2 3` tại $t=4 \rightarrow$ `-1`
- T6: bật lại 3 \rightarrow `? 2 3` tại $t=6 \rightarrow$ độ không ổn định là: 6 (gốc) + 5 (nút 2) + 1 (nút 3 được bật lại tại $t=5$) = 14

Bạn có muốn mình tiếp tục viết luôn giải pháp chi tiết cho bài này với cấu trúc cây và thuật toán tối ưu không?