

Bài A. ALIENGIFT

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

Người ngoài hành tinh vừa đến trái đất, đậu ngay trước sân nhà Hùng. Họ đưa cho anh ta n hộp quà tặng, đánh số từ 1 đến n . Trong hộp quà tặng thứ i có một lò xo có độ cứng có thể thay đổi được trong khoảng $[L_i, H_i]$ và 4 món quà được đánh số 1,2,3,4. Mỗi chìa khóa là một chuỗi có 4 bit. Khi bạn sử dụng một chìa khóa nào đó để mở hộp, bạn sẽ nhận được một số món quà trong hộp ứng với các vị trí có bit 1 trên chìa khóa. Ví dụ: Với chìa khóa 0110, bạn sẽ có món quà 2 và 3

Nếu bạn đang đứng ở hộp thứ i , bạn sẽ sử dụng một chìa khóa nào đó để mở hộp và lấy các món quà, sau đó bạn sử dụng lò xo để nhảy đến hộp khác. Nếu bạn sử dụng lò xo với độ cứng k ($k \in [L_i, H_i]$) bạn sẽ nhảy đến hộp $i+k$, hoặc ra khỏi sân nếu $i+k > n$. Theo yêu cầu của người ngoài hành tinh, chìa khóa không được chứa 2 bit 1 liên tiếp, và 2 chìa khóa dùng liên tiếp nhau thì không được có bit 1 ở cùng vị trí. VD bạn vừa sử dụng chìa 0010 thì việc sử dụng chìa 1010 ở bước tiếp theo sẽ không hợp lệ.

Các món quà có giá trị là một số nguyên, Hùng đang đứng ở hộp quà tặng 1, anh ta muốn sử dụng các lò xo và chìa khóa một cách hợp lệ để thu được nhiều giá trị quà tặng nhất. Tất nhiên, việc không chọn món quà nào có vẻ không hiểu khách và là không hợp lệ. Anh ta sẽ không dừng lại cho đến khi nhảy ra khỏi sân

Dữ liệu vào

- Dòng đầu chứa số nguyên n
- n dòng tiếp theo, dòng thứ i chứa 6 số nguyên: L_i H_i và giá trị của 4 món quà ở hộp thứ i

Kết quả

- Một số nguyên duy nhất là giá trị tối đa có thể thu thập

Ví dụ

stdin	stdout
6 1 1 3 2 -4 5 1 2 2 -3 1 3 2 2 1 1 -1 -1 1 3 -10 10 30 33 1 1 2 3 -5 4 1 100 2 2 2 2	59

Hạn chế

- $n \leq 10^5$. $0 < L_i \leq H_i \leq 10^9$. |Giá trị các món quà| $\leq 10^9$
- Có 50% số test với $n \leq 1000$

Bài B. XSUM

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

Ánh đang cố xây dựng một ma trận $n \times m$ như sau:

- Ban đầu ma trận còn trống, được chia làm $n \times m$ ô vuông đơn vị
- Ánh sẽ điền lần lượt các số vào các ô theo thứ tự từ trên xuống dưới, từ trái sang phải
- Khi điền đến ô (i, j) , cô sẽ tìm số tự nhiên nhỏ nhất chưa xuất hiện trên dòng i cũng như trên cột j và điền số này vào ô (i, j)

Sau khi điền xong, Ánh nhận thấy ma trận này rất đặc biệt và muốn tính tổng các phần tử của nó.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n ($n \leq 10^{1000}$)
- Dòng thứ hai chứa số nguyên dương m ($m \leq 10^{1000}$)

Kết quả

Ghi tổng các số trên ma trận, sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

stdin	stdout
3 3	12
4 4	24

Bài C. RADIO

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

Năm 2050, loài người có bước đột phá trong việc chinh phục vũ trụ. Họ xây dựng được n trạm không gian thẳng hàng, cách đều nhau $1s$ ánh sáng. Các trạm không gian được đánh số từ 1 đến n , xuất phát từ trạm 1 là trái đất, theo thứ tự xa dần trái đất (các trạm nằm về cùng 1 phía so với trái đất)

Do đặc điểm về cách mã hóa thông tin của các trạm, trạm không gian thứ i có khả năng nhận tín hiệu radio từ các trạm có số hiệu Ll_i đến Lr_i ($1 \leq Ll_i \leq Lr_i \leq i$), và gửi tín hiệu radio đến các trạm có số hiệu từ Rl_i đến Rr_i ($i \leq Rl_i \leq Rr_i \leq n$). Tốc độ gửi đi của sóng radio trong không gian là tốc độ ánh sáng. Thời gian cần thiết để trạm thứ i xử lý tín hiệu (giải mã tín hiệu nhận được thành thông tin hoặc mã hóa thông tin gửi đi thành tín hiệu radio) là $t_i(s)$. Điều này có nghĩa, để có thể gửi thông tin từ trạm i sang trạm j ($i < j$, $Ll_j \leq i \leq Lr_j$ và $Rl_i \leq j \leq Rr_i$) thì trạm i cần mã hóa thông tin, gửi tín hiệu sau khi mã hóa sang trạm j , và giải mã tín hiệu nhận được ở trạm j , do đó mất thời gian $t_i + j - i + t_j(s)$

Yêu cầu: Tính toán thời gian ít nhất để gửi thông tin từ trái đất đến các trạm khác

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên: n
- n dòng tiếp theo, dòng thứ i chứa 5 số nguyên: $t_i \ Ll_i \ Lr_i \ Rl_i \ Rr_i$

Kết quả

Ghi $n - 1$ số nguyên trên 1 dòng, là thời gian cần thiết để gửi thông tin từ trái đất đến trạm $2, 3, \dots, n$. Với những trạm không thể gửi thông tin đến thì ghi -1

Ví dụ

stdin	stdout
5	10 5 7 11
2 1 1 2 4	
7 1 1 5 5	
1 1 2 4 5	
0 2 3 4 5	
3 1 4 5 5	

Hạn chế

- $1 \leq n \leq 10^5$, $0 \leq t_i \leq 10^9$ trong tất cả các test
- 20% với $1 \leq n \leq 5000$
- 20% với $Ll_i = 1$, $Lr_i = i$ với mọi $i = 1, 2, \dots, n$

Bài D. XGAME

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

Cho một cây có gốc có n đỉnh, các đỉnh được đánh số từ 1 đến n với gốc là đỉnh 1. Xét một trò chơi trên cây này như sau:

- Có hai người, luân phiên nhau thực hiện lượt chơi
- Đến lượt mình, người chơi chọn một cạnh của cây và xóa đi. Khi đó, toàn bộ cây con ứng với cạnh đó sẽ bị xóa (tức là những đỉnh không liên thông với gốc đều sẽ bị xóa)
- Ai không thực hiện được lượt chơi hợp lệ nữa sẽ thua cuộc. Rõ ràng là trò chơi sẽ kết thúc sau hữu hạn bước, nên sẽ không có kết quả hòa

Bạn sẽ chơi trò này với máy, bạn được quyền chọn người đi trước và hãy dành chiến thắng trong trò chơi.

Tương tác:

- Đầu tiên bạn cần đọc vào số n ($0 \leq n \leq 5000$)
- Tiếp theo bạn cần đọc vào $n - 1$ số, số thứ i là đỉnh cha của đỉnh $i + 1$
- Sau đó bạn cần in ra 1 hoặc 0 tương ứng là bạn muốn đi trước hoặc đi sau
- Sau đó trò chơi sẽ bắt đầu. Khi đến lượt máy, máy sẽ in ra một số và bạn cần phải đọc vào số này và chuyển sang lượt chơi của bạn. Khi đến lượt bạn, bạn cần in ra một số và chuyển sang lượt chơi của máy
- Số được người chơi in ra là chỉ số của đỉnh xa gốc hơn trong số hai đỉnh của cạnh muốn xóa (tức là nếu bạn in ra x thì bạn sẽ xóa cạnh nối x với cha của x)
- Trò chơi sẽ kết thúc khi không có cạnh nào có thể đến được từ gốc nữa. Lúc này bạn cần kết thúc chương trình của mình (không đọc vào cũng không in ra gì nữa)

Lưu ý, sau mỗi lần in ra bạn cần đẩy dữ liệu ra luồng chuẩn (`flush(stdout)` hoặc `cout << endl`) để tương tác được với máy.

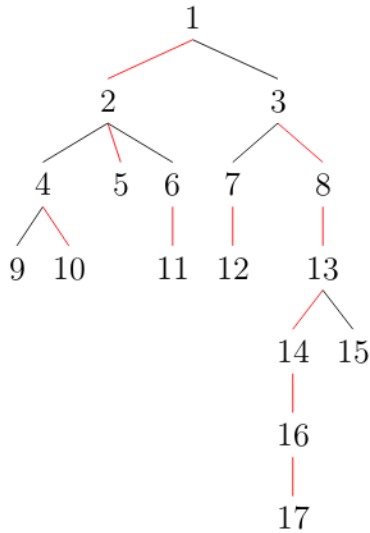
Ví dụ

<code>stdin</code>	<code>stdout</code>
8	1
1 1 2 2 3 3 4	8
4	5
3	2

Bài E. HLD

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

Cho một cây có gốc gồm n đỉnh. Các đỉnh được đánh số từ 1 đến n với gốc là đỉnh 1. Với mỗi đỉnh x không phải là lá, cần chọn một đỉnh y con của x và đánh dấu cạnh xy là cạnh nặng. Những cạnh không được đánh dấu gọi là cạnh nhẹ. Khi giải quyết các bài toán truy vấn trên cây, các đoạn cạnh nặng liên tiếp nhau thường được quản lý bằng cây quản lý phạm vi để giảm độ phức tạp tính toán.



Trong bài này, ta sẽ định nghĩa chi phí của một cách chọn cạnh nặng như sau: Với x là một nút trên cây, xét đường đi từ x lên gốc. Chia đường đi này thành các đoạn liên tiếp sao cho các cạnh trong một đoạn thì cùng loại và số đoạn là ít nhất có thể (tức là đường đi sẽ được chia thành các đoạn cạnh nặng liên tiếp và các đoạn cạnh nhẹ liên tiếp, xen kẽ nhau). Chi phí tính toán của một đoạn cạnh nặng có L cạnh là $\text{ceil}(\log_2(L))+1$ và của một đoạn cạnh nhẹ có L cạnh là L . Chi phí tính toán của nút x là tổng chi phí tính toán của tất cả các đoạn cạnh nói trên. Chi phí của một cách chọn cạnh nặng là chi phí tính toán lớn nhất trong số các chi phí tính toán của các nút trên cây. Có thể hiểu chi phí của cách chọn cạnh nặng như là độ phức tạp tính toán của một truy vấn trong trường hợp xấu nhất

Yêu cầu: Tìm chi phí nhỏ nhất có thể có của một cách chọn cạnh nặng.

Dữ liệu vào

- Dòng đầu tiên chứa số lượng testcase: T ($1 \leq T \leq 10$)
- Dòng đầu tiên của mỗi testcase chứa một số nguyên dương là số đỉnh của cây: n ($1 \leq n \leq 10^5$)
- $n - 1$ dòng tiếp theo mỗi dòng chứa hai số nguyên dương mô tả một cạnh của cây: $u \ v$

Kết quả

Với mỗi testcase, in ra trên một dòng chi phí nhỏ nhất có thể có của một cách chọn cạnh nặng

Ví dụ

stdin	stdout
3	4
17	3
1 2	1
1 3	
2 4	
2 5	
2 6	
3 7	
3 8	
4 9	
4 10	
6 11	
7 12	
8 13	
13 14	
13 15	
14 16	
16 17	
12	
1 2	
2 4	
4 5	
5 6	
4 7	
1 3	
3 8	
8 9	
3 10	
10 11	
10 12	
2	
1 2	