

Bài 1. (6 điểm) Bánh sinh nhật

Nhân ngày sinh nhật của thầy Hoàng, các học sinh CSP tổ chức một bữa tiệc chúc mừng. Bánh trái được bày trên một bàn dài chạy nối từ phòng đội tuyển ra tận cổng chính Xuân Thủy. Đặc biệt là có n bánh ngọt, chiếc bánh thứ i được đặt ở vị trí x_i tính từ đầu bàn ở phòng đội tuyển. Thầy rất ngạc nhiên và cảm động trước thịnh tình của các học sinh nên tuyên bố sẽ cố gắng ăn càng nhiều bánh ngọt càng tốt để không phụ công mọi người trong việc chuẩn bị. Tuy nhiên, hôm nay thầy cũng không có quá nhiều thời gian. Bà xã thầy cũng chuẩn bị tiệc ở nhà nên giao cho thầy một việc vô cùng quan trọng là ... đón con! Tính đi tính lại, thầy chỉ có thể dành T đơn vị thời gian để ăn uống với học sinh. Rất may, Thiên Anh – một chuyên gia ăn uống đã nhanh chóng tính được giúp thầy các thông số về chiếc bánh thứ i (ở vị trí x_i , với $i > j$ thì $x_i \geq x_j$) để ăn hết sẽ cần t_i thời gian. Để đi từ vị trí i đến vị trí j cần $|x_i - x_j|$ thời gian. Ở cùng một vị trí có thể có nhiều bánh, việc di chuyển là không cần thiết nhưng bánh thì phải ăn lần lượt từng chiếc.

Yêu cầu: Xuất phát từ vị trí 0, hãy xác định số bánh nhiều nhất mà thầy Hoàng có thể ăn trong khoảng thời gian T .

Dữ liệu: Vào từ file **CAKE.INP**:

- Dòng đầu tiên chứa 2 số nguyên n và T ($1 \leq n \leq 10^5$, $1 \leq T \leq 10^9$),
- Dòng thứ i trong n dòng sau chứa 2 số nguyên x_i t_i ($1 \leq x_i, t_i \leq 10^9$).

Kết quả: Đưa ra file văn bản CAKE.OUT số bánh tối đa thầy Hoàng có thể được ăn.

Ví dụ:

CAKE . INP	CAKE . OUT
3 10 1 4 2 5 3 3	2

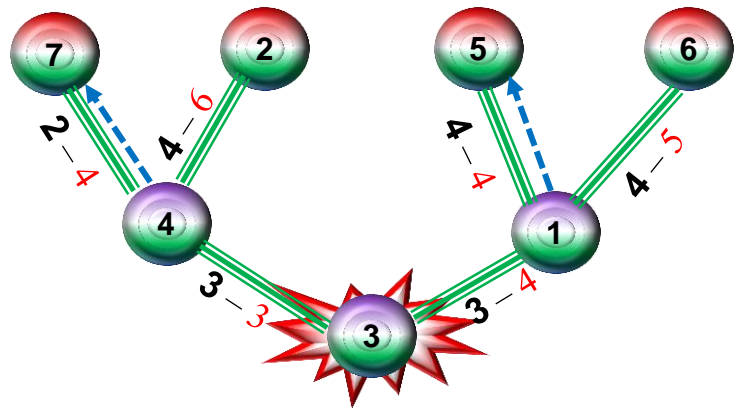
Ràng buộc:

- 50% test $n \leq 1000, T \leq 10^9$
- 50% test $n \leq 10^5, T \leq 10^9$

2. Sơ tán

Một tổ hợp nghiên cứu đại dương có n phòng đặt ngầm dưới nước và có $n-1$ hành lang hai chiều đảm bảo việc di chuyển giữa tất cả các phòng. Nói một cách khác Tổ hợp có dạng cây mà mỗi phòng là một nút và hành lang là cạnh. Các phòng chỉ có một hành lang nối có phần nhô ra ngoài mặt nước và có xuồng cứu hộ. Các phòng này được gọi là phòng nổi.

Chi phí xây dựng và duy trì hoạt động của tổ hợp là rất lớn vì vậy việc đảm bảo an toàn hệ thống trước các thiên tai được khảo sát và tính toán kỹ ngay ở giai đoạn thiết kế. Kết quả mô phỏng trên máy tính cho thấy nếu có động đất xảy ra, đường hành lang thứ i độ dài d_i nối 2 phòng a_i và b_i có thể chịu đựng được t_i đơn vị thời gian, sau đó sẽ bị sụp.



Khi có báo động các nhân viên phải mau

chống di chuyển về phòng nổi để sơ tán khỏi tổ hợp. Thời gian di chuyển qua hành lang đúng bằng độ dài của hành lang đó. Thời gian băng qua một phòng là không đáng kể. Nếu đang di chuyển trong một hành lang và hành lang đó bị sụp thì không thể cứu được. Hành lang bị sụp dĩ nhiên sẽ không đi qua được nữa. Nếu di chuyển tới phòng mới đúng lúc hành lang sụp thì vẫn an toàn. Có thể người ở một số phòng không kịp thoát tới phòng nổi. Những phòng này thiết kế lại để các nhân viên làm việc trong đó vẫn an toàn, ở lại tại chỗ chờ cứu hộ đặc biệt tới sau.

Yêu cầu: Hãy xác định theo thiết kế hiện tại có bao nhiêu phòng không đòi hỏi phải thiết kế lại.

Dữ liệu: Vào từ file văn bản EVACUATION.INP:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$)
- Dòng thứ i trong $n-1$ dòng sau chứa 4 số nguyên dương a_i, b_i, d_i, t_i ($d_i \leq 10^4, t_i \leq 10^9$).

Kết quả: Đưa ra file văn bản EVACUATION.OUT một số nguyên – số lượng phòng không đòi hỏi phải thiết kế lại.

EVACUATION . INP	EVACUATION . OUT
7	6
1 6 4 5	
3 4 3 3	
5 1 4 4	
7 4 2 4	
2 4 4 6	
3 1 3 4	

Ví dụ:

50% test $n \leq 1000$

3. BÁNH KẸP

Cửa hàng *Snow Crab* có món bánh mì kẹp hải sản cực kỳ nổi tiếng. Tiêu chí phục vụ của cửa hàng là trong ngày không có hai khách hàng nào nhận được bánh giống nhau. Các món đặt vào bánh kẹp được đánh giá phân loại bởi chỉ số nguyên không âm a_1, a_2, a_3, \dots . Nếu bánh được kẹp các món với chỉ số a_1, a_2, \dots, a_m thì chỉ số của bánh thành phẩm sẽ là $a_1 \oplus a_2 \oplus \dots \oplus a_m$, trong đó \oplus là phép tính **xor** (phép ^ trong C++). Các món được chuẩn bị cho ngày nào thì chỉ dùng trong ngày đó và phải *dùng hết trong ngày*.

Nhân viên cơ quan, công sở không còn đi tới đi tới các cửa hàng Com trưa văn phòng, thay vào đó họ gửi yêu cầu tới *Snow Crab* và được mang bánh tới tận tay.

Hôm nay cửa hàng nhận được k yêu cầu, yêu cầu thứ i muốn có bánh với chỉ số không nhỏ hơn u_i và không lớn hơn v_i , $i = 1 \div k$. Bộ phận chế biến cung cấp n món. Các món được băng chuyền chuyển sang bộ phận kẹp bánh đóng gói thành phẩm. Bộ phận đóng gói sẽ lấy ra dãy các món liên tục (tính từ đầu theo trình tự xuất hiện), đóng thành một chiếc bánh kẹp, cho vào hộp và chuyển tới địa chỉ đặt hàng. Tất cả đều được tự động hóa, nhưng bếp trưởng muốn biết với các món đã chuẩn bị thì có nhiều cách khác nhau đáp ứng yêu cầu của khách hàng hay không để không phải chuẩn bị thêm các món loại khác. Các yêu cầu được đáp ứng theo trình tự gửi tới nhà hàng.

Hãy đưa ra số lượng cách đóng gói khác nhau có thể thực hiện. Hai cách đóng gói gọi là khác nhau nếu tồn tại ít nhất một món thuộc một bánh trong cách đóng gói thứ nhất và thuộc bánh khác trong cách đóng gói thứ hai.

Dữ liệu: Vào từ file văn bản BURGER.INP:

- ✚ Dòng đầu tiên chứa 2 số nguyên n và k ,
- ✚ Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$, $i = 1 \div n$),
- ✚ Dòng thứ j trong k dòng sau chứa 2 số nguyên u_j và v_j ($0 \leq u_j \leq v_j \leq 10^9$).

Kết quả: Đưa ra file văn bản BURGER.OUT số lượng cách đóng gói khác nhau có thể thực hiện theo mô đun 10^9+7 .

Ví dụ:

BURGER.INP	BURGER.OUT
7 3	6
1 0 1 0 1 0 1	
1 1	
0 0	
1 1	

20% test $n \leq 100, k \leq 4$

40% test tiếp theo $n \leq 500, k \leq 100$

40% test còn lại ($1 \leq n \times k \leq 10^5, k \leq n$)

