

VQ45. MUA HÀNG TRỰC TUYẾN

Khi được nuôi nhiều để phục vụ các thí nghiệm khoa học và điều chế vắc xin. Ban quản lý đảo khi được cung cấp ngân sách để mua chuối làm thức ăn bổ sung cho khi. Ngân sách được rút về tài khoản của Ban quản lý từng phần và được chia thành n lần, lần thứ i số tiền a_i được chuyển về tại thời điểm t_i . Ban đầu, số dư tài khoản là 0.

Thông tin trên mạng cho thấy có m công ty có thể cung cấp hàng. Công ty thứ j có hàng và nhận đặt ở thời điểm u_j . Nếu đặt hàng thì lô hàng sẽ được mang tới vào thời điểm v_j , $j = 1 \div m$. Với mỗi lô hàng bên mua có thể thanh toán trực tuyến vào thời điểm đặt hàng với giá $c1$, khách hàng cũng có thể trả sau vào thời điểm hàng được mang tới với giá $c2$ ($c1 \leq c2$). Hàng chỉ có thể mua được khi có đủ tiền thanh toán ở một trong 2 thời điểm nói trên. Do không biết trước thời điểm tiền được chuyển vào tài khoản nên mọi yêu cầu chào hàng đều được đặt. Nếu đến thời điểm nhận mà số dư tài khoản vẫn không đủ thanh toán thì lô hàng sẽ bị trả về.

Hãy xác định số lượng lô hàng nhiều nhất mà Ban quản lý đảo có thể mua, biết rằng không có thời điểm nào xuất hiện nhiều hơn một nơi có thể đặt hàng, không có 2 đơn hàng nào được mang tới cùng một lúc và không có có thời điểm đặt hàng nào trùng với thời điểm mang hàng tới của công ty khác. Nói một cách hình thức, với $p \neq q$ có $u_p \neq u_q$, $v_p \neq v_q$ và $u_p \neq u_q$.

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

- ✚ Dòng đầu tiên chứa 2 số nguyên $c1$ và $c2$ ($1 \leq c1 \leq c2 \leq 1\,000$),
- ✚ Dòng thứ 2 chứa số nguyên n ($1 \leq n \leq 10^5$),
- ✚ Dòng thứ i trong n dòng sau chứa 2 số nguyên a_i và t_i ($1 \leq a_i \leq 1\,000$, $1 \leq t_i \leq 10^9$),
- ✚ Dòng tiếp theo chứa số nguyên m ($1 \leq m \leq 10^5$),
- ✚ Dòng thứ j trong m dòng sau chứa 2 số nguyên u_j và v_j ($1 \leq u_j \leq v_j \leq 10^9$).

Kết quả: Đưa ra file văn bản BESTBUY.OUT một số nguyên – số lượng lô hàng nhiều nhất mà Ban quản lý đảo có thể mua.

Ví dụ:

BESTBUY.INP	BESTBUY.OUT
100 200	3
3	
100 1	
200 10	
400 21	
4	
12 22	
2 4	
5 23	
8 19	

VQ42. 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 Cơm 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 bằng chuyển 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 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 ($1 \leq n \times k \leq 10^5$, $k \leq n$),
-  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						
7	3					
1	0	1	0	1	0	1
1	1					
0	0					
1	1					

BURGER.OUT
6