



09

BÀI TẬP TỔNG HỢP



1☀. Tìm số

Cho số nguyên dương N và P . Tìm N số dương a_1, a_2, \dots, a_N sao cho:

$a_1 \times a_2 \times \dots \times a_N = P$ và $GCD(a_1, a_2, \dots, a_N)$ đạt giá trị lớn nhất.

Dữ liệu cho trong file **FindNum.Inp** gồm hai số nguyên dương N và P .

Kết quả ghi ra file **FindNum.Out** là giá trị lớn nhất của $GCD(a_1, a_2, \dots, a_N)$.

Ví dụ:

FindNum.Inp	FindNum.Out	Giải thích
3 24	2	Ta có dãy: 2, 2, 6.

Giới hạn: $1 \leq N, P \leq 10^{12}$.



2☀. Dãy bit

Cho chuỗi S chỉ gồm các kí tự thuộc $\{0, 1\}$. Một phép biến đổi là chọn đúng m kí tự liên tiếp và đảo m kí tự đó: từ kí tự 1 chuyển thành kí tự 0 và từ kí tự 0 thành kí tự 1.

Yêu cầu: Tìm một số phép biến đổi ít nhất để biến chuỗi S thành chuỗi toàn các kí tự giống nhau.

Dữ liệu vào từ file BITSTR.INP gồm:

- Dòng 1 chứa chuỗi S độ dài không quá 10^6 chỉ gồm các kí tự $\{0, 1\}$.
- Dòng 2 chứa số nguyên dương $m \leq 10^6$.

Kết quả ghi ra file BITSTR.OUT là một số nguyên duy nhất là số phép biến đổi cần thực hiện, nếu không thể thực hiện phép biến đổi thì ghi -1.

Ví dụ:

BITSTR.INP	BITSTR.OUT
1110111	2
4	



3☀. Mạng rút gọn.

Hệ thống giao thông của nước XYZ có n thành phố và m con đường hai chiều nối giữa hai thành phố. Các thành phố được đánh số từ 1 đến n . Chuẩn bị đón chào ngày quốc khánh, Bộ giao thông nước XYZ tiến hành khảo sát độ *tốt* của đường và phân các đường ra làm ba loại: Loại 1, loại 2 và loại 3. Các con đường loại 1 và loại 2 chỉ cho xe máy đi qua, các con đường loại 2 và loại 3 chỉ cho ô tô đi qua. Bộ giao thông muốn chọn một số nhiều nhất các con đường để sửa nhưng các con



đường còn lại luôn đảm bảo sự thông suốt đối với cả hai loại xe, tức là xe máy có thể đi được giữa hai thành phố bất kì và ô tô cũng có thể đi được giữa hai thành phố bất kì.

Dữ liệu cho trong file MRG.INP như sau:

- Dòng đầu ghi hai số nguyên dương n, m ($n \leq 10000; m \leq 50000$).
- M dòng tiếp theo, mỗi dòng ghi ba số nguyên dương u, v, k mô tả đường đi từ u đến v , k là loại của con đường ($1 \leq k \leq 3$), không có hai con đường nào cùng nối một cặp thành phố.

Kết quả ghi trong file MRG.OUT gồm một số duy nhất là số con đường nhiều nhất mà Bộ giao thông có thể sửa (nếu không có cánh nào để liên thông thì ghi -1).

Ví dụ:

MRG.INP	MRG.OUT	Hình minh họa
<pre> 6 9 1 2 1 1 3 3 2 3 2 2 4 1 2 5 2 3 5 3 4 5 2 4 6 1 5 6 3 </pre>	2	



4. Đếm cặp

Cho đồ thị gồm n đỉnh được đánh số từ 1 đến n , đỉnh thứ i có màu c_i . Người ta thêm lần lượt m cạnh vô hướng vào đồ thị, cạnh thứ j nối hai đỉnh u_j và v_j .

Yêu cầu: Sau mỗi bước thêm cạnh, đếm số cặp đỉnh i, j cùng màu mà từ i có thể đến j qua các cạnh của đồ thị ($i < j$).

Dữ liệu: cho trong file VCPAIRS.INP gồm:

- Dòng 1 ghi hai số nguyên dương n và m ($n \leq 10^5, m \leq 2 \cdot 10^5$).
- Dòng 2 ghi n số nguyên dương c_1, c_2, \dots, c_n ($c_i \leq 10^9$).
- m dòng cuối, dòng thứ j ghi hai số nguyên u_j và v_j .

Kết quả: Ghi ra file VCPAIRS.OUT gồm m dòng, mỗi dòng ghi số cặp đỉnh i, j cùng màu mà từ i có thể đi đến j qua các cạnh của đồ thị ($i < j$).

Ví dụ:

VCPAIRS.INP	VCPAIRS.OUT
4 4	0
1 2 1 2	0
1 2	2
3 4	2
1 3	
2 3	



5☀. Những chuyến bay

John cần phải đi từ thành phố A đến thành phố B. Cho biết có n chuyến bay gồm các thông tin sau: x là thành phố xuất phát, y là thành phố đích, s là thời điểm xuất phát, t là thời điểm đến.

Thời gian làm thủ tục để đi một chuyến bay là p , nghĩa là nếu thời điểm có mặt tại thành phố x là t_l mà muốn đi chuyến bay $\langle x, y, s, t \rangle$ thì $t_l + p < s$ (và sẽ có mặt ở y tại thời điểm t).

Yêu cầu: Cho biết r là thời điểm John đang có mặt ở thành phố A, tìm thời điểm sớm nhất mà John có thể đến B.

Dữ liệu cho trong file **Airplane.inp** gồm:

- Dòng đầu ghi 5 số nguyên A, B, n, r, p ;
- n dòng sau, mỗi dòng gồm 4 số nguyên x_i, y_i, s_i, t_i .

Kết quả ghi ra file **Airplane.Out** là thời điểm sớm nhất mà John có thể đến B.

Ví dụ:

Airplane.inp	Airplane.Out
111 222 5 5 10	270
111 222 1 101	
111 222 15 100	
111 222 100 300	
111 333 20 160	
333 222 171 270	

Giới hạn:

- Sub1: $n \leq 1000$; $0 < A, B, r, p \leq 1000$; $x_i, y_i, s_i, t_i \leq 1000$;
- Sub2: $n \leq 100000$; $0 < A, B, r, p \leq 10^9$; $x_i, y_i, s_i, t_i \leq 10^9$;



