



Ngôn ngữ lập trình C++

\$01\$ BÀI TẬP LẬP TRÌNH NÂNG CAO



1☀. Rectangle 0-1

Cho hình chữ nhật gồm R dòng và C cột. Trong các ô vuông nhỏ có ghi số 0 hoặc số 1. Ta có thể trao đổi bất kì hai cột nào cho nhau và số lần trao đổi là tùy ý.

Yêu cầu: Tìm hình chữ nhật chỉ gồm số 1 (tất cả các số trong hình chữ nhật cần tìm đều bằng 1) có kích thước lớn nhất (kích thước bằng tích số dòng nhân với số cột của hình chữ nhật đó).

Dữ liệu cho trong file REC01.INP như sau:

- Dòng đầu ghi hai số nguyên dương R và C .
- R dòng sau, mỗi dòng ghi C số 0 hoặc 1.

Kết quả ghi ra file REC01.OUT là kích thước của hình chữ nhật lớn nhất tìm được.

Ví dụ:

REC01.INP	REC01.OUT
3 5 0 1 0 1 0 0 1 0 1 1 1 1 0 1 0	6

Giới hạn:

- 20% test : $R, C \leq 100$;
- 30% test : $R, C \leq 500$;
- 50% test : $R, C \leq 1000$;



2☀. Lò sưởi

Bảo tàng Mùa đông ở đất nước XYY nổi tiếng với những tư liệu quý, hiện vật có tuổi đời hàng trăm năm. Các tư liệu và hiện vật đều gắn với những sự kiện lịch sử của đất nước và nền văn minh nhân loại. Vì vậy bảo tàng Mùa đông là một điểm tham quan mà nhiều khách du lịch tới thăm khi đến đất nước XYY. Một ngày đẹp trời nhưng nhiệt độ vẫn lạnh, bảo tàng có n khách tham quan đến thăm. Thời điểm khách thứ i ($i = 1, 2, \dots, n$) đến thăm là T_i và rời khỏi bảo tàng vào thời điểm $T_i + 1$ ($1 \leq T_1 < T_2 < \dots < T_n \leq 10^9$).

Để giữ ấm cho khách tham quan, bảo tàng sử dụng một chiếc lò sưởi. Điều đặc biệt của chiếc lò sưởi là không được bật (tắt) quá k lần. Nhân viên điều khiển lò sưởi muốn tìm cách bật/tắt lò sưởi sao cho:

- Số lần bật/tắt lò sưởi không quá k lần.
- Trong tất cả thời gian mà các khách tới thăm thì lò sưởi đều ở trạng thái hoạt động.
- Tổng số thời gian mà lò sưởi hoạt động là ít nhất.

Dữ liệu cho trong file STOVE.INP gồm:

- Dòng đầu ghi hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).



- Dòng thứ i trong n dòng tiếp theo, ghi số nguyên dương T_i .

Kết quả ghi ra file STOVE.OUT là tổng thời gian hoạt động ít nhất của lò sưởi.

Ví dụ:

STOVE.INP	STOVE.OUT	Giải thích
3 2 1 3 6	4	- Bật lò sưởi từ thời điểm 1 đến 4 (thời gian hoạt động $4 - 1 = 3$). - Bật lần thứ 2 từ thời điểm 6 đến 7 (thời gian hoạt động $7 - 6 = 1$). Tổng thời gian hoạt động ít nhất là 4.
3 1 1 2 6	6	Lò sưởi được bật từ thời điểm 1 đến thời điểm 7. Thời gian hoạt động bằng $7 - 1 = 6$.



3☀. Hạ cánh – (Duyên hải lớp 10 - 2020)

Sau khi khống chế thành công COVID19, các đường bay đã được mở lại, nhu cầu đi lại tăng cao sau kì nghỉ Tết dài nhất trong lịch sử. Hiện tại là thời điểm 0, có N máy bay đang tiếp cận để hạ cánh tại sân bay Cát Bi. Máy bay thứ i ($1 \leq i \leq N$) có thể điều chỉnh tốc độ để hạ cánh ở một mốc thời điểm nguyên trong khoảng thời gian $[L_i, R_i]$. Trong đó L_i là thời điểm sớm nhất máy bay có thể hạ cánh, R_i là thời điểm muộn nhất máy bay có thể hạ cánh, quá thời R_i , máy bay sẽ chuyển hướng hạ cánh tại sân bay khác. Khoảng thời gian $R_i - L_i$ được gọi là giới hạn chờ của máy bay thứ i và giới hạn này tất cả N máy bay là giống nhau.

Sân bay có K đường băng, có thể hoạt động độc lập và tiếp nhận các máy bay hạ cánh. Các máy bay phải thực hiện lệnh giãn cách X giây. Hai máy bay liên tiếp hạ cánh trên một đường băng phải cách nhau ít nhất X giây.

Yêu cầu: Hãy lên phương án sắp xếp các máy bay, sao cho số lượng máy bay hạ cánh là nhiều nhất có thể. Nếu có cùng phương án đảm bảo số lượng máy bay hạ cánh nhiều nhất, tìm phương án tối ưu sao cho thời gian chênh lệch nhỏ nhất giữa hai máy bay cùng hạ cánh trên một đường bay là lớn nhất.

Dữ liệu: vào từ file LANDING.INP gồm:

- Dòng đầu ghi 3 số nguyên dương N, K, X ($N \leq 10^5, K \leq 4, X \leq 10^9$).
- Tiếp theo ghi N dòng, mỗi dòng ghi 2 số nguyên L_i, R_i ($0 \leq L_i \leq R_i \leq 10^9$).

Dữ liệu đảm bảo $R_i - L_i$ của tất cả máy bay là giống nhau.

Kết quả ghi ra file LANDING.OUT gồm hai số T và P , trong đó:

T là số máy bay nhiều nhất có thể hạ cánh được.

P là chênh lệch nhỏ nhất giữa thời điểm hạ cánh của hai máy bay trên một đường băng.

Nếu mọi đường băng đều có nhiều nhất 1 máy bay thì $P = -1$.

Ví dụ:



LANDING.INP	LANDING.OUT	Giải thích
5 2 60 0 20 0 20 100 120 60 80 110 130	5 65	Đường băng 1: • MB 1 thời điểm 0 • MB 4 thời điểm 65 • MB 5 thời điểm 130 Đường băng 2: • MB 2 thời điểm 0 • MB 3 thời điểm 100



4. Hành trình du lịch

Đất nước xinh đẹp XYY có N điểm du lịch nổi tiếng. Hệ thống đường cao tốc gồm $N - 1$ con đường nối trực tiếp giữa các điểm du lịch. Mỗi con đường nối trực tiếp một cặp điểm du lịch sao cho giữa hai điểm du lịch bất kì luôn có đường đi giữa chúng (có thể là con đường trực tiếp hoặc qua những con đường nối với nhau). Các điểm du lịch được đánh số hiệu $1, 2, \dots, N$. Các con đường cho phép di chuyển cả hai chiều.

Một khách du lịch hiện đang ở điểm du lịch u và muốn đi thăm tất cả các điểm du lịch u_1, u_2, \dots, u_k .

Cho biết độ dài của mỗi con đường cao tốc. Hãy tìm cho khách du lịch một hành trình xuất phát từ điểm du lịch u và đi thăm tất cả các điểm du lịch u_1, u_2, \dots, u_k với tổng độ dài cần di chuyển là ít nhất.

Dữ liệu cho trong file **TourU.Inp** gồm:

- Dòng 1 ghi số nguyên dương N là số địa điểm du lịch.
- $N - 1$ dòng tiếp theo, mỗi dòng ghi ba số nguyên dương a, b, c mô tả tuyến đường cao tốc nối điểm du lịch a với b và có độ dài là c ($1 \leq a \neq b \leq N; c \leq 10^6$);
- Dòng tiếp theo ghi số nguyên dương u là điểm hiện khách du lịch đang đứng.
- Dòng tiếp sau là số nguyên k là số địa điểm cần đến thăm.
- Dòng cuối ghi k số nguyên khác nhau u_1, u_2, \dots, u_k mô tả k điểm cần đến thăm của khách du lịch.

Chú ý là u khác với u_1, u_2, \dots, u_k .

Kết quả ghi ra file **TourU.Out** là tổng độ dài cần di chuyển là ít nhất.

Ví dụ:



TourU.Inp	TourU.Out	Hình minh họa
10 1 2 1 2 3 2 3 4 3 4 5 1 3 6 3 3 7 6 6 8 5 8 9 3 8 10 4 2 4 3 5 6 7	22	

Giải thích: Hành trình di chuyển tối ưu: Xuất phát từ đỉnh 2:

- $2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 3 \rightarrow 7$:
- Tổng khoảng cách di chuyển: $2 + 3 + 1 + 1 + 3 + 3 + 3 + 6 = 22$.



5☀️. Virus (VOI 2017)

“TextFile” là một virus chuyên tấn công các file văn bản theo phương thức sau: Chép một đoạn các kí tự liên tiếp trong nội dung của file văn bản vào bộ nhớ trong, thay đổi một số ký tự trong đoạn này, sau đó chèn đoạn văn bản đã thay đổi và ngay sau đoạn văn bản vừa sao chép trong file văn bản. Vinh đang phát triển phần mềm để phát hiện một file văn bản đã bị nhiễm virus nói trên hay chưa. Vì thế, Vinh cần giải quyết bài toán sau: Cho một chuỗi T và số nguyên không âm k , chuỗi con gồm các kí tự từ vị trí i đến j của chuỗi T được gọi là đoạn có khả năng bị virus sao chép mức k nếu nó sai khác với chuỗi con gồm các kí tự liên tiếp từ vị trí $j + 1$ đến vị trí $j + (j - i + 1)$ của chuỗi T ở không quá k vị trí.

Ví dụ: $T = \text{'xubaa'}$, đoạn văn bản ‘**ab**’ từ vị trí 2 đến 3 là đoạn văn bản độ dài 2 có khả năng bị virus sao chép mức 1 vì nó khác với đoạn văn bản ‘**aa**’ tại 1 vị trí. Tất nhiên đoạn văn bản ‘**ab**’ cũng là đoạn có khả năng bị rút sao chép mức 2, 3, ...

Yêu cầu: Cho chuỗi T và n số nguyên không âm k_1, k_2, \dots, k_n . Với mỗi giá trị k_i , tìm độ dài đoạn dài nhất trong chuỗi T có khả năng bị sao chép mức k_i ($i = 1, 2, \dots, n$).

Dữ liệu vào từ file VIRUS.INP gồm:

- Dòng đầu chứa số nguyên dương n ($n \leq 10$).
- Dòng thứ hai chứa một chuỗi T gồm các chữ cái latin thường.
- Dòng thứ i trong n dòng tiếp theo ghi số nguyên k_i ($k_i \leq 1000, i = 1, 2, \dots, n$).

Kết quả ghi ra file VIRUS.OUT gồm n dòng, dòng thứ i ghi một số nguyên không âm là độ dài của đoạn dài nhất có khả năng bị virus sao chép mức k_i ($i = 1, 2, \dots, n$). Ghi 0 nếu không tìm được đoạn như vậy.

Ví dụ:



VIRUS.INP	VIRUS.OUT
2	1
xabaaxy	2
0	
1	
2	1
xabaabxc	3
0	
1	

Giới hạn:

- Đặt $m = \text{length}(T)$.
- Có 40% test với $m \leq 300$
- Có thêm 30% test với $m \leq 3000$, $k_i = 0$ với $i = 1, 2, \dots, n$.
- 30% số test còn lại, $m \leq 3000$.



6. Dãy nhị phân độ dài K.

Cho hai số nguyên dương n và k . Hãy đếm số các dãy nhị phân độ dài n sao cho trong k bit liên tiếp trong dãy luôn có ít nhất một bit 1.

Dữ liệu vào từ file **BinaryK.Inp** gồm hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^6$).

Kết quả ghi ra file **BinaryK.Out** là $S \% (10^9 + 7)$ trong đó S là số các dãy nhị phân độ dài n , thỏa mãn trong k bit liên tiếp của dãy luôn có ít nhất một bit 1.

Ví dụ:

BinaryK.Inp	BinaryK.Out	Giải thích
3 2	5	Có 5 dãy: 010, 011, 101, 110, 111.