



Ngôn ngữ lập trình C++

§02§ BÀI TẬP LẬP TRÌNH NÂNG CAO



1☀. Sắp xếp theo bit

Cho số nguyên dương N . Hãy sắp xếp các số $1, 2, 3, \dots, N$ theo tiêu chí sau:

- Các bit 1 khi biểu diễn ở hệ nhị phân tăng dần.
- Nếu các bit 1 bằng nhau thì giá trị tăng dần.

Dữ liệu cho trong file BitSort.Inp gồm số nguyên dương N ($N \leq 10^6$).

Kết quả ghi ra file BitSort.Out gồm N dòng ghi N số trong dãy được sắp xếp

Ví dụ:

BitSort.Inp	BitSort.Out
4	1 2 4 3



2☀. Dãy DSP

Một dãy số nguyên dương a_1, a_2, \dots, a_k được gọi là dãy DSP nếu a_{j+1} là ước của $a_1 + a_2 + \dots + a_j$ ($j = 1, 2, \dots, k - 1$).

Yêu cầu: Cho dãy x_1, x_2, \dots, x_n , hãy giữ lại nhiều phần tử nhất trong dãy, sau đó sắp xếp lại để nhận được dãy DSP.

Dữ liệu vào từ file DSP.inp:

- Dòng đầu chứa số nguyên n ($0 < n \leq 20$);
- Dòng thứ hai chứa n số nguyên dương x_1, x_2, \dots, x_n ($x_i \leq 10^9$).

Kết quả ghi ra file DSP.out:

- Dòng đầu chứa số nguyên k là số phần tử được giữ lại;
- Dòng thứ hai gồm k số nguyên dương là chỉ số các phần tử được giữ lại, các chỉ số được liệt kê theo thứ tự các phần tử tương ứng tạo thành dãy DSP.

Ví dụ:

DSP.inp	DSP.out
3	3
1 2 2	2 3 1

Ràng buộc:

Subtask 1: $n \leq 10$; [50% số điểm]

Subtask 2: $10 < n \leq 20$; [50% số điểm]

**3. 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: $2 + 3 + 1 + 1 + 3 + 3 + 3 + 6 = 22$.



4. 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.