

Câu 1: Dãy con tăng trọng số (WISEQ)

Xét dãy số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$). Một dãy chỉ số $1 \leq i_1 < i_2 < \dots < i_k \leq n$ mà $a_{i_1} < a_{i_2} < \dots < a_{i_k}$ thì dãy $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ được gọi là một dãy con tăng của dãy a_1, a_2, \dots, a_n với độ dài của dãy là k và tổng $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ được gọi là trọng số của dãy con tăng.

Yêu cầu: Cho dãy số nguyên dương a_1, a_2, \dots, a_n và số nguyên dương W , hãy tìm dãy con tăng của dãy a_1, a_2, \dots, a_n có độ dài lớn nhất và trọng số không vượt quá W .

Dữ liệu: Vào từ file văn bản **WISEQ.INP**

- Dòng đầu chứa hai số nguyên dương n, W ($n \leq 50000$)
- Dòng thứ hai gồm n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$)

Kết quả: Đưa ra file văn bản **WISEQ.OUT** gồm một dòng chứa một số nguyên là độ dài dãy con tăng lớn nhất tìm được thỏa mãn yêu cầu.

Ví dụ

WISEQ.INP
5 10
4 2 3 1 5
5 5
4 2 3 1 5

WISEQ.OUT
3
2

Giới hạn: Đặt $S = a_1 + a_2 + \dots + a_n$

- Subtask 1: $n \leq 20$; $W \leq S$;
- Subtask 2: $n \leq 50$; $W = S$;
- Subtask 3: $n \leq 50$; $W \leq S$;
- Subtask 4: $n \leq 500$; $W \leq S$;
- Subtask 5: $n \leq 50000$; $W = S$;

Câu 2: Chi phí nhỏ nhất (MINCOST)

Sau khi giải được bài toán Dãy con tăng trọng số, Steve nhận được phần thưởng là một chuyến du lịch tới một đất nước Alpha vô cùng xinh đẹp. Đất nước này gồm có N thành phố và M con đường hai chiều kết nối các thành phố với nhau. Điều kì lạ ở đất nước này là nếu Steve muốn đi qua một con đường nào đó, cậu phải đưa ra cho người quản lí con đường đó xem một số đồng xu với đủ mệnh giá khác nhau. Lưu ý rằng Steve chỉ cần cho họ xem những đồng xu đó thôi nên mỗi đồng xu có thể được sử dụng lại ở nhiều con đường khác nhau. Đương nhiên, trước đó cậu sẽ phải đổi tiền của mình lấy những đồng xu của đất nước này rồi, mệnh giá tiền xu ở đây cũng rất đặc biệt: có K mệnh giá tiền xu c_1, c_2, \dots, c_K trong đó luôn có $2c_{i-1} \leq c_i$ ($2 \leq i \leq K$).

Steve muốn xuất phát tại một thành phố bất kì và đi thăm thú tất cả N thành phố của đất nước Alpha nhưng cũng không muốn phải đổi quá nhiều tiền, nên cần tính toán tổng mệnh giá các đồng xu phải đổi nhỏ nhất để dù xuất phát ở thành phố nào cũng có thể đi thăm tất cả N thành phố.

Yêu cầu: Cho trước thông tin N thành phố, M con đường và mệnh giá các đồng xu cần có để đi qua từng con đường. Hãy giúp Steve xác định tổng mệnh giá các đồng xu phải đổi nhỏ nhất để Steve có thể đi thăm tất cả N thành phố.

Dữ liệu: Vào từ file văn bản **MINCOST.INP**

- Dòng đầu tiên gồm 3 số nguyên dương N, M, K ($N, M, K \leq 10^5$);
- Dòng thứ 2 gồm K số nguyên c_1, c_2, \dots, c_K là các mệnh giá của K đồng xu;
- Dòng thứ i trong M dòng tiếp theo: ba số nguyên dương đầu tiên u_i, v_i, t_i với u_i, v_i là số hiệu hai thành phố, t_i là số lượng đồng xu mà Steve phải đưa cho người quản lí xem khi đi qua con đường nối giữa thành phố u_i và v_i , t_i số tiếp theo là các số nguyên dương biểu diễn thứ tự của các đồng xu trong dãy K đồng xu.

Kết quả: Đưa ra file văn bản **MINCOST.OUT** gồm một dòng ghi một số nguyên duy nhất là đáp án bài toán. Nếu không tồn tại cách đổi tiền thỏa mãn, in ra -1.

Ví dụ:

MINCOST.INP
3 3 4
1 2 5 10
1 2 2 1 2
1 3 1 3
2 3 1 4

MINCOST.OUT
8

Giải thích: Chọn các đồng xu thứ 1, 2, 3 có tổng mệnh giá là $1 + 2 + 5 = 8$, ta có thể thăm tất cả 3 thành phố bằng cách đi qua các con đường nối thành phố 1 và 2, 1 và 3.

Giới hạn:

- Subtask 1: 30% số test có $N, M \leq 2000, 1 \leq c_i \leq 1000$.
- Subtask 2: 30% số test có $N, M \leq 10^5, 1 \leq c_i \leq 1000$.
- Subtask 3: 40% số test có $N, M \leq 10^5, 1 \leq c_i \leq 10^{18}$.

Câu 3: Bàn phím (KEYBOARD)

Sau khi đi du lịch từ thành phố Alpha về, Steve vẫn còn dư tiền thưởng và quyết định đi mua bàn phím mới để phục vụ cho công việc gõ mật khẩu. Mật khẩu là một xâu s có độ dài là n . Mỗi kí tự của xâu này là một trong m kí tự đầu tiên trong bảng chữ cái Latin in

thường. Bàn phím thể hệ mới là một hoán vị của m chữ cái Latin in thường đầu tiên được xếp theo một hàng ngang từ trái sang phải.

Ví dụ, nếu $m = 3$ thì có 6 bàn phím khác nhau là: abc, acb, bac, bca, cab và cba . Vì Steve chỉ gõ mật khẩu với một ngón tay nên cần thời gian để di chuyển từ một kí tự trong mật khẩu sang kí tự tiếp nối nó. Thời gian Steve dùng để di chuyển từ kí tự s_i sang kí tự s_{i+1} bằng với khoảng cách của hai kí tự này trên bàn phím. Thời gian tổng cộng mà Steve cần để gõ mật khẩu s với một bàn phím nhất định thì được gọi là độ chậm của bàn phím đó. Steve muốn mua bàn phím có độ chậm nhỏ nhất.

Định nghĩa cụ thể hơn, độ chậm của một bàn phím thì bằng

$$\sum_{i=2}^n |pos(s_{i-1}) - pos(s_i)|$$

Với $pos(x)$ là vị trí của kí tự x trên bàn phím.

Ví dụ, nếu s là $aacabc$ và bàn phím là bac , thì tổng thời gian để gõ mật khẩu là:
 $|pos(a) - pos(a)| + |pos(a) - pos(c)| + |pos(c) - pos(a)| + |pos(a) - pos(b)| + |pos(b) - pos(c)| = |2 - 2| + |2 - 3| + |3 - 2| + |2 - 1| + |1 - 3| = 0 + 1 + 1 + 1 + 2 = 5$

Yêu cầu: Cho trước chuỗi kí tự s là mật khẩu Steve hay phải gõ, hãy giúp Steve chọn mua bàn phím có độ chậm nhỏ nhất.

Dữ liệu: Vào từ file văn bản **KEYBOARD.INP**

- Dòng đầu tiên chứa hai số nguyên dương n, m ($1 \leq n \leq 10^5; 1 \leq m \leq 20$).
- Dòng thứ hai chứa một chuỗi s có n kí tự. Mỗi kí tự là một trong m chữ cái Latin đầu tiên (in thường).

Kết quả: Đưa ra file văn bản **KEYBOARD.OUT** in ra một số nguyên là độ chậm nhỏ nhất mà một bàn phím có thể có.

Ví dụ

KEYBOARD.INP	KEYBOARD.OUT
6 3 aacabc	5
15 4 abacabadabacaba	16

Giải thích: Ví dụ đầu tiên được xem xét trong đề bài. Trong ví dụ thứ hai, bàn phím tốt nhất là $bacd$.

Ghi chú:

- Subtask 1: 10% số test có $0 < M \leq 2$;
- Subtask 2: 30% số test có $2 < M \leq 10$;
- Subtask 3: 60% số test có $10 < M \leq 20$.