



ĐỀ CHÍNH THỨC

(Đề thi gồm 04 trang)

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File chương trình	Input/Output	Điểm
1	Tập số	opset.cpp	Thiết bị chuẩn	100
2	Mật khẩu	pw.cpp	Thiết bị chuẩn	100
3	Mạng công ty	comnet.cpp	Thiết bị chuẩn	100
4	Mê cung ngoặc	bmzgame.cpp	Thiết bị chuẩn	100

Bài 1. Tập số

Trên tập số $\{1, 2, \dots, n\}$, Alice tiến hành xóa đi k ($k < n$) số $a_1, a_2, \dots, a_k$ để nhận được tập S . Một cách chọn các số trên tập S được gọi là cách chọn tối ưu bậc d nếu:

- Hiệu hai số bất kì được chọn có giá trị tuyệt đối lớn hơn d ;
- Số lượng số được chọn là lớn nhất.

Ví dụ, trên tập số $\{1, 2, 3, 4, 5, 6, 7, 8\}$, xóa đi ba số 2, 3, 8 được tập $S = \{1, 4, 5, 6, 7\}$, ba tập $\{1, 4, 6\}$, $\{1, 4, 7\}$ và $\{1, 5, 7\}$ đều là cách chọn tối ưu bậc 1.

Yêu cầu: Cho n, k, d và dãy $a_1, a_2, \dots, a_k$, hãy giúp Alice tính số lượng số chọn được trong cách chọn tối ưu bậc d và số cách chọn tối ưu. Chú ý, hai cách chọn được gọi là khác nhau nếu tồn tại một số của S thuộc trong cách chọn này nhưng không thuộc trong cách chọn kia.

Dữ liệu: Vào từ thiết bị vào chuẩn (bàn phím) có khuôn dạng:

- Dòng đầu chứa ba số nguyên dương n, k, d ($k < n \leq 10^7; k \leq 10^5; d \leq n$);
- Dòng thứ hai chứa k số nguyên dương phân biệt $a_1, a_2, \dots, a_k$ ($a_i \leq n, 1 \leq i \leq k$);

Kết quả: Ghi ra thiết bị ra chuẩn (màn hình) hai dòng, dòng thứ nhất là số lượng số chọn được trong cách chọn tối ưu, dòng thứ hai là số cách chọn tối ưu chia dư cho $(10^9 + 7)$.

Input	Output
8 3 1	3
2 3 8	3

Subtask 1 (20 điểm): $n - k \leq 20$;

Subtask 2 (25 điểm): $n - k \leq 200$;

Subtask 3 (25 điểm): $n - k \leq 2 \times 10^5$;

Subtask 4 (30 điểm): Không có ràng buộc gì thêm.

Bài 2. Mật khẩu

Alice muốn đặt mật khẩu cho một ứng dụng mà cô mới xây dựng. Cô đã chọn chuỗi ký tự S (kí hiệu $|S|$ là độ dài chuỗi S) và dự định chọn K đoạn trên chuỗi S (các đoạn gồm ít nhất một ký tự và không nhất thiết rời nhau) rồi ghép các đoạn theo một thứ tự nào đó để nhận được chuỗi đối xứng. Nhắc lại, chuỗi đối xứng là chuỗi đọc từ trái qua phải cũng như đọc từ phải qua trái, ví dụ “*abba*”, “*sos*” là chuỗi đối xứng, còn chuỗi “*abab*” thì không phải là chuỗi đối xứng. Alice đã chọn $K - 1$ đoạn, đoạn thứ i ($1 \leq i < K$) gồm các ký tự thứ L_i đến ký tự thứ R_i của chuỗi S ($1 \leq L_i \leq R_i \leq |S|$). Khi chọn đoạn thứ K , Alice muốn chọn một đoạn có độ dài m mà với đoạn đó Alice có thể ghép với $K - 1$ đoạn đã chọn theo một thứ tự nào đó để nhận được một chuỗi đối xứng.

Yêu cầu: Cho chuỗi S và $(K - 1)$ cặp số L_i, R_i , hãy đếm số cách chọn đoạn thỏa mãn.

Dữ liệu: Vào từ thiết bị vào chuẩn (bàn phím) có khuôn dạng:

- Dòng đầu chứa hai số nguyên dương K, m ($m \leq |S|$);
- Dòng thứ hai chứa chuỗi S chỉ gồm các ký tự ‘a’ đến ‘z’ ($2^K \times |S| \leq 2 \times 10^5$);
- Dòng thứ i ($1 \leq i \leq K - 1$) trong $K - 1$ dòng tiếp theo chứa hai số nguyên dương L_i, R_i ($1 \leq L_i \leq R_i \leq |S|$).

Kết quả: Ghi ra thiết bị ra chuẩn (màn hình) một dòng chứa một số là số lượng cách chọn thỏa mãn.

Input	Output
1 1 abab	4

Input	Output
2 2 abab 2 3	2

Subtask 1 (20 điểm): $K = 1; |S| \leq 2000$;

Subtask 2 (20 điểm): $K = 1$;

Subtask 3 (20 điểm): $K \leq 7; |S| \leq 2000$;

Subtask 4 (20 điểm): $K \leq 7$.

Subtask 5 (20 điểm): $K = 14$.

Bài 3. Mạng công ty

Công ty của Alice có n máy tính, các máy tính được đánh số từ 1 đến n . Hiện tại đang có m ($m \geq n - 1$) dây nối giữa các máy tính, dây nối thứ k ($1 \leq k \leq m$) nối hai máy tính u_k, v_k ($u_k \neq v_k$) và giúp truyền tin theo cả hai chiều giữa hai máy, có thể có nhiều dây nối giữa hai máy tính. Hiện tại, n máy tính có thể không liên thông với nhau, Alice có thể tháo dây nối để đấu nối lại với mong muốn làm cho n máy tính liên thông, cụ thể, cô có thể thực hiện:

- Tháo một đầu nối của dây thứ k để đầu nối sang máy tính khác, hành động này mất chi phí c_k ;
- Tháo cả hai đầu nối của dây thứ k để đầu nối sang hai máy tính khác, hành động này mất chi phí $2 \times c_k$.

Yêu cầu: Hãy giúp Alice tính chi phí ít nhất cần thực hiện để liên thông được n máy tính.

Dữ liệu: Vào từ thiết bị vào chuẩn (bàn phím) có khuôn dạng:

- Dòng đầu chứa hai số nguyên dương n, m ($n \leq 10^5; n - 1 \leq m \leq 2 \times 10^5$);
- Dòng thứ k ($1 \leq k \leq m$) trong m dòng tiếp theo chứa ba số nguyên dương u_k, v_k, c_k ($c_k \leq 10^6$).

Kết quả: Ghi ra thiết bị ra chuẩn (màn hình) một dòng chứa một số là chi phí ít nhất tìm được.

Input	Output
3 3 1 2 1 1 2 2 1 3 1	0

Input	Output
3 3 1 2 1 1 2 2 1 2 3	1

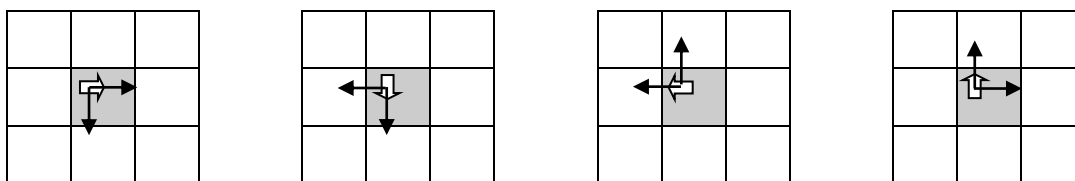
Subtask 1 (50 điểm): $c_i = 1$;

Subtask 2 (25 điểm): $m, n \leq 1000$;

Subtask 3 (25 điểm): Không có ràng buộc gì thêm.

Bài 4. Mê cung ngoặc

Một mê cung được mô tả bằng bảng chữ hình chữ nhật kích thước $m \times n$. Các hàng của bảng được đánh số từ 1 đến m , từ trên xuống dưới, các cột của bảng được đánh số từ 1 đến n , từ trái qua phải. Ô nằm trên giao của hàng i và cột j được gọi là ô (i, j) . Mỗi ô của lưới chứa một kí tự ngoặc mở '(' hoặc ngoặc đóng ')'



Người chơi sẽ xuất phát từ ô $(1, 1)$, quay hướng tới phía ô $(1, n)$ và di chuyển trên bảng. Việc di chuyển phải tuân thủ các quy tắc được mô tả trong hình trên, cụ thể: từ ô đang đứng, căn cứ vào hướng đang hướng tới được chỉ ra bởi mũi tên $\rightarrow$, thực hiện bước di chuyển sang ô kề cạnh đang hướng tới, hoặc sang ô kề cạnh nằm bên phải (các hướng có thể di chuyển được chỉ ra bởi các mũi tên $\rightarrow$). Mỗi ô chỉ được đi qua nhiều nhất một lần. Người chơi có thể dừng di chuyển tại một ô nào đó để kết thúc trò chơi.

Khi kết thúc trò chơi, người chơi nhận được một xâu kí tự T gồm các kí tự trong các ô trên đường đi được xếp liên tiếp nhau. Người chơi giành chiến thắng nếu xâu T là một biểu thức ngoặc đúng bậc k .

Nhắc lại, định nghĩa biểu thức ngoặc đúng và bậc của biểu thức ngoặc.

- Biểu thức rỗng là biểu thức ngoặc đúng và có bậc bằng 0,
- Nếu A là biểu thức ngoặc đúng có bậc bằng k thì (A) cũng là một biểu thức ngoặc đúng có bậc bằng $k + 1$,
- Nếu A và B là hai biểu thức ngoặc đúng và có bậc tương ứng là k_1 và k_2 thì AB cũng là một biểu thức ngoặc đúng có bậc bằng $\max(k_1, k_2)$.

Ví dụ, $'()()'$ là một biểu thức ngoặc đúng có bậc bằng 2 còn $'()()()'$ là một biểu thức ngoặc đúng và có bậc bằng 3.

Yêu cầu: Cho bảng chữ và số nguyên dương k , đếm số lượng đường đi khác nhau giúp người chơi giành chiến thắng. Hai đường đi được gọi là khác nhau nếu tồn tại một ô thuộc đường đi này nhưng không thuộc đường đi kia.

Dữ liệu: Vào từ thiết bị vào chuẩn (bàn phím) có khuôn dạng:

- Dòng đầu tiên ghi ba số nguyên dương m, n, k ($m, n \leq 30; k \leq 10$);
- Tiếp theo là m dòng mô tả bảng chữ, mỗi dòng chứa một xâu gồm n kí tự, mỗi kí tự ngoặc mở '(' hoặc ngoặc đóng ')'

Kết quả: Ghi ra thiết bị ra chuẩn (màn hình) một dòng là số lượng đường đi đếm được chia dư cho $(10^9 + 7)$.

Input	Output
3 3 1 ())) ())))	4

Subtask 1 (50 điểm): $m, n \leq 5$;

Subtask 2 (25 điểm): $k = 1$;

Subtask 3 (25 điểm): Không có ràng buộc gì thêm.

----- **HẾT** -----

Họ và tên thí sinh: Số báo danh:

Lưu ý:

- Thí sinh **không** được sử dụng tài liệu.
- Cán bộ coi thi **không** giải thích gì thêm.