Bài 9: Em đã đánh cắp trái tim anh – STOLENHEART (1.0s, 8G)

“Ma trận trái tim anh đã bị em đánh cắp, anh chẳng còn “sống” nữa nếu thiếu em, nhưng em muốn +1 kill nên anh đành chấp nhận.”

An – nhân vật đã xuất hiện trong Day 1 quyết định nghiên cứu ma trận trái tim của Anh Coder May Mắn vì:

- Anh muốn biết tại sao Anh Coder May Mắn lại có thể chịu đựng cú đá của người yêu cũ.

- An muốn hoàn thành luận văn tiến sĩ ở Viện Nghiên cứu Tim 4.0 với đề tài: “Nghiên cứu sự suy thoái trái tim của một người từng lãnh trọn cú đá tinh thần”.

- Sau tất cả, An muốn tìm một người có hoàn cảnh giống mình để có thể chia sẻ với nhau về chuyện đời.

-Và hơn hết, An rảnh.

Có thể biểu diễn trái tim của Anh Coder May Mắn thành T ma trận, mỗi ma trận có kích thước M x N. Các ô trong ma trận là những ô thuộc các loại sau:

. là ô trống

E*x* là ô lan truyền cảm xúc mạnh cấp x. Cảm xúc mạnh cấp x có thể lan ra như sau: Ô đó: cảm xúc cấp x. 8 ô xung quanh: cảm xúc cấp x-1. Những ô tiếp xúc (có tính hướng chéo) với 8 ô xung quanh ô E*x*: cảm xúc cấp x-2. Và cứ lan ra như thế cho đến khi cảm xúc được ảnh hưởng bởi ô đó bằng 0.

S*x* là ô diệt trừ cảm xúc cấp x. Nguyên tắc cũng giống ô cảm xúc nhưng thay vì cộng cảm xúc, nó trừ đi cảm xúc. Và cảm xúc không bao giờ là một giá trị <0.

W là ô vô cảm. Nếu được tác động bởi ô E*x* hay S*x* P lần thì nó sẽ biến 4 ô chung quanh nó thành ô có cảm xúc 0, cố định không bao giờ thay đổi và 2 ô hướng chéo trên thành ô E*p* còn hai ô hướng chéo dưới thành ô S*p* . Mặc định cảm xúc của ô vô cảm là 0. Nếu có ô nào đã được sửa chữa thì ô vô cảm sẽ không tác động đến ô đó.

? là những ô An cần phải sửa.

*x* là những ô không phải là ô lan truyền cảm xúc, nhưng nó có thể ảnh hưởng đến những ô cảm xúc khác như sau:

Nếu có 8 ô cảm xúc cấp *x* bao quanh ô cảm xúc cấp *y* với (y < x) thì ô đó sẽ tăng K đơn vị cảm xúc (Lưu ý: một khi đã tăng đơn vị cảm xúc thì ô đó sẽ không bao giờ tăng đơn vị cảm xúc nữa.)

Input: Dòng đầu: số T, thể hiện số testcase

Mỗi testcase có cách thức như sau:

Dòng đầu: Bốn số M, N, P, K

M dòng sau, mỗi dòng N kí tự như trên

Output: Với mỗi testcase, in ra giá trị cảm xúc của tất cả các ô trong ma trận theo thứ tự từ trên xuống dưới, trái qua phải ở dòng đầu tiên và in ra ô cảm xúc có giá trị lớn nhất và nhỏ nhất trong ma trận ở dòng thứ hai (Nếu có ô nào không thể phục hồi, gán giá trị NaN)

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  3 3 1 0  E5 ? ?  ? ? ?  ? ? ? | 5 4 3 4 4 3 3 3 3  5 3 |

Subtasks:

Subtask 1 (3%) : M, N < 100, P, K < 500, T < 30. Không có ô vô cảm và ô S*x*.

Subtask 2 (17%) : 100 < M, N < 1000000, 500 < P, K < , 30 < T < 10000 . Không có ô vô cảm.

Subtask 3 (30%) ; 1000000 < M, N < , < P, K < , 10000 < T <

Subtask 4 (50%) : < M, N < , < P, K < , < T <

Sau cùng, An cũng phục hồi được gần như toàn bộ ma trận trái tim của Anh Coder May Mắn.  
Các ô cảm xúc đã lấp lại – theo quy luật, theo dòng, theo cột.  
Những vị trí trống được tính toán từ trung bình, đối xứng, hoặc tổng tuyến tính.  
Tất cả… trừ đúng một ô.

Ô (r, c) – nơi cảm xúc ban đầu là “không rõ”.  
Dù thử bao nhiêu hàm restore(x, y),  
Dù chạy mọi tổ hợp dòng và cột,  
An vẫn chỉ nhận được giá trị NaN.

Cậu ghi vào báo cáo:

*"Trái tim đã được phục hồi 99.99%.  
Riêng một ô – mã hóa ban đầu không rõ, không thể tính ngược."*

An không hiểu vì sao mình thấy khó chịu với điều đó.  
Cậu không quen cảm xúc. Cậu chỉ quen biến.  
Nhưng khi đóng file heart\_matrix\_final.out,  
cậu nhìn lại ô (r, c) lần nữa…

…và gõ vào một dòng chú thích:

**"Có thể ô này từng mang một giá trị quan trọng. Chỉ số cảm xúc của ô đó có thể đã là .  
Nhưng cậu ấy đã xoá nó từ trước khi tôi bắt đầu phục hồi."**