TỔNG HỢP BÀI DAY 2

Bài 7: Lật mặt 7 – FLIPBIT (1.0s, 8G)

Hết chuyện của An, giờ chúng ta sẽ đến với chuyện của “Anh Coder May Mắn”. Anh là một người bị “đá” còn nhiều hơn cả An, lên tới lần! Nhưng anh ấy là một người rất thông minh, nên đã nghĩ ra một trò chơi để “test” độ thông minh của người yêu. Anh ấy cho T lưới bit M x N, mỗi ô là 1 bit 0 hoặc 1. Ở tọa độ (x, y) (0-indexed) sẽ xuất hiện một con “kiến”. Cách thức hoạt động của nó như sau:

- Nếu dưới chân nó là ô 0, nó sẽ biến ô đó thành 1 sau đó đi sang phải nhưng vẫn giữ nguyên hướng cũ nếu số bit 1 là lẻ (không tính bit ở dưới vị trí con kiến), xoay và đi sang trái không giữ nguyên hướng cũ nếu số bit 1 là chẵn.

- Nếu dưới chân nó là ô 1, nó sẽ biến ô đó thành ô 0 sau đó xoay sang trái nếu có số lượng bit 1 là lẻ và xoay sang phải nếu có số lượng bit 1 là chẵn. Sau đó đi thẳng.

- Nếu bị kẹt, hủy tất cả các truy vấn còn lại và in ra lưới ngay lập tức.

Hãy trả lời P truy vấn các loại, và những kiểu truy vấn như sau:

P: In lưới.

PR *x*: In hàng x

PC *x*: In cột x

PB *x y*: In ô (x, y)

S *x*: Cho con kiến chạy thêm x bước nữa.

Cuối cùng, hãy in ra lưới sau tất cả truy vấn trên.

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

Mỗi testcase có cách thức như sau:

Dòng đầu: Sáu biến M, N, X, Y, P, *ch* – thể hiện hướng của con kiến.

M dòng sau, mỗi dòng N kí tự thể hiện 0, 1.

P dòng cuối, mỗi dòng một truy vấn.

Output: Với mỗi testcase, in ra:

- Hàng, cột, ô hay cả lưới nếu được yêu cầu (truy vấn kiểu P)

- Cuối cùng, in ra cả lưới sau tất cả các truy vấn

- Dòng cuối: in ra hướng của con kiến.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  5 5 2 1 2 ^  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  S 2  PC 1 | 0 0 1 0 0  0 0 0 0 0  0 0 0 0 0  1 1 0 0 0  0 0 0 0 0  0 0 0 0 0  < |

Subtasks:

Subtask 1 (5%) : M, N < 100, P < 30

Subtask 2 (15%) : 100 < M, N < 1000000, 30 < P < 500.

Subtask 3 (30%) ; 1000000 < M, N < , 500 < P <

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

“Cuộc đời anh coder “may mắn” cũng giống như con kiến này vậy, quẹo trái rồi lại quẹo phải mà chẳng thể tìm một điểm dừng chân cho bản thân. Anh ấy tự thắc mắc: “Liệu có ai sẵn lòng fix hết đống bug đời mình không?””

Bài 8: Anh nhảy việc nuôi em còn em nhảy người yêu nhanh hơn tốc độ giãn nở của ngân hà – TURTLENINJA (1.0s, 8G)

“Lời “cuối cùng nổi tiếng” của Anh Coder May Mắn trước khi chính thức gia nhập hội Michelin Boy lần thứ : Tại sao, lúc nào tôi cũng bị đá?”

Lý do chính khiến anh ấy bị đá lên tới lần là do… quá nghiện game. Anh ấy có thể speedrun mọi loại game với tốc độ cực nhanh: Phá đảo Minecraft trong vòng 30s, lập luôn kỉ lục thế giới (!?), chơi Candy Crush Saga nhiều đến mức đội ngũ ý tưởng còn không kịp tạo màn mới vì chỉ cần 0.1s thì anh ấy sẽ phá đảo hết. Tuy vậy, Anh Coder May Mắn viết code C++ lại không được clean cho lắm (?!) do vậy cái công việc của anh cũng chẳng thể giữ thêm bao lâu nữa.

“Tình cờ” hôm nay, anh ấy phát hiện ra một con game mới toanh trên Steam có tên là: TURTLE NINJA. Mục tiêu trong game này như sau:

Ninja “rùa” phải di chuyển đến đích sao cho tổng chi phí là nhỏ nhất. Biết Ninja Rùa có thể di chuyển như sau:

|  |  |
| --- | --- |
| x o . o x  o . . . o  . . N . .  o . . . o  x o . o x | N: vị trí của Ninja Rùa  o: Ninja Rùa có thể “nhảy” đến vị trí đó.  x: Ninja Rùa có thể “đi’ đến vị trí đó. (Không thể nhảy) |

Chi phí di chuyển theo đường chéo ban đầu là D, tiếp theo là D + A, sau nữa là D + 2A, cứ thế cộng dồn. Còn chi phí di chuyển theo đường thẳng ban đầu là S, tiếp theo là S + C, sau đó là S + 2C, cứ thế cộng dồn.

Trò chơi này có L màn chơi, mỗi màn là một lưới M x N riêng biệt gồm các loại ô sau:

. là ô trống

W là ô tường. Ô tường có tác dụng “cản” cách di chuyển theo đường chéo và “chặn” điểm đến làm giới hạn Ninja Rùa.

N là vị trí của ninja. Chỉ có duy nhất 1 kí tự N trong mỗi màn chơi.

F là ô đích đến. Chỉ có duy nhất 1 kí tự F trong mỗi màn chơi.

P*x* là cổng dịch chuyển. Khi đến ô này, Ninja Rùa có thể (không bắt buộc) dịch chuyển tới một ô dạng P*x* khác với chi phí Q.

<kí tự biểu thị hướng di chuyển>*x* là một ô làm cho Ninja Rùa nhảy theo hướng được chỉ định x ô. Ví dụ >5 có nghĩa là Ninja Rùa sẽ nhảy đến một ô cách 5 ô so với ô gốc theo hướng bên phải. Nếu không thể nhảy x ô vì giới hạn màn chơi, nhảy đến ô xa nhất theo hướng đó nằm trong màn chơi.

Với mỗi màn chơi, hãy giúp Anh Coder May Mắn tính toán chi phí tối thiểu.

Input: Dòng đầu: số L, thể hiện số màn chơi

Mỗi testcase có cách thức như sau:

Dòng đầu: Bảy biến D, A, S, C, M, N, Q

M dòng sau, mỗi dòng N kí tự như trên.

Output: In ra L dòng, dòng thứ i là chi phí tối thiểu để vượt qua màn i.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  1 1 1 1 4 4 2  N . . .  . . . .  . . F .  . . . . | 1 |

Subtasks:

Subtask 1 (5%) : M, N < 100, D, A, S, C, Q < , L < 1000, không có ô cổng dịch chuyển và ô di chuyển tự động.

Subtask 2 (15%) : 100 < M, N < 1000000, < D, A, S, C, Q < , 1000 < L < , không có ô cổng dịch chuyển.

Subtask 3 (30%) ; 1000000 < M, N < , < D, A, S, C, Q < , < L < ,

Subtask 4 (50%) : < M, N < , < D, A, S, C, Q < , < L <

“Đường tình duyên của anh ấy như cách mà chú Ninja Rùa này di chuyển:  
Rõ ràng ở ngay trước mắt...  
Nhưng lại phải đi ngoằn ngoèo, vòng vèo qua biết bao chướng ngại,  
Tốn không biết bao nhiêu chi phí, thời gian, cả... tình cảm.  
Và khi đến được chỗ em,  
Em đã tay trong tay với người khác rồi.”

“**Recommended.**

Game khá ổn, chỉ tiếc là lúc đến được đích thì bạn gái tôi đã cưới người khác. 4.5/5. Chờ phần 2. – một anh chàng coder “nào đó”.”

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."**

Bài 10: Hành Trình Không Dừng – NONSTOPJOURNEY (1.0s, 4G)

Sau khi chữa tim cho Anh Coder May Mắn và bảo vệ thành công luận án tiến sĩ, hai người quyết định cày rank TBCP chế độ 2v2 cùng nhau. Họ phải thắng chính xác W trận mới có thể lên mức rank “Thách Đấu Người Ra Đề TBCP”. Thể lệ thi đấu 2v2 như sau:

Ngân hàng đề bài của TBCP mỗi lần vào trận gồm N bài, bài thứ i có độ khó . Mục tiêu của hai người là chọn những bài có độ khó ít nhất và phù hợp với trình độ của mỗi người (An nếu gặp một bài có độ khó H mod K = P, ngay lập tức giảm M độ khó còn Anh Coder May Mắn nếu gặp một bài có độ khó H chính bằng với X là số nguyên, ngay lập tức độ khó giảm thành làm tròn xuống của ). Quy tắc chọn bài trên TBCP cũng giống như quy tắc chọn tướng của Liên Minh Huyền Thoại.

Cả phần chọn bài gồm R vòng đấu, mỗi vòng có thể là vòng Cấm (B) hoặc Chọn (P). Vòng cấm là khi hai bên cấm những đề bài mà họ cho rằng là dễ. (Đối thủ biết cả hai người có năng lực như đã kể trên nhưng họ không có năng lực tương tự.) Vòng chọn là khi cả hai người chọn một đề bài mà họ cho rằng là dễ nhất. Phần chọn bài được biểu diễn dưới dạng 1 xâu chỉ gồm các kí tự B và P có độ dài R. Đội của An và Anh Coder May Mắn đi trước vào vòng cấm/chọn đầu tiên, sau đó đến vòng thứ hai, đội đối thủ sẽ cấm/chọn trước, cứ thế luân phiên.

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

Mỗi testcase có cách thức như sau:

Dòng đầu: Bảy biến N, K, P, M, A, Q, R.

Dòng sau: Dãy thể hiện độ khó của các bài.

Dòng cuối: Xâu thể hiện phần chọn bài

Output: Với mỗi testcase, in ra như sau:

Dòng đầu: Độ khó tối thiểu mà họ chọn được

Dòng sau: Số thứ tự những bài họ đã chọn (dựa theo dãy nhập ở đầu, 1-indexed)

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  2 23456 789 9999 1234 123456 1  3 1  P | 1  2 |

Subtasks:

Subtask 1 (3%) : N < 100, K, P, M, A, Q < 10000, R < 10, W < 100, chỉ có lượt chọn

Subtask 2 (15%) : 100 < N < 1000000, 10000 < K, P, M, A, Q < , 10 < R < 1000, 100 < W < 100000

Subtask 3 (30%) ; 1000000 < N < , < K, P, M, A, Q < , 1000 < R < , 100000 < W <

Subtask 4 (52%) : < N < , < K, P, M, A, Q < , < R < , < W <

“An không lòng vòng, gửi thẳng một link: play.tbcp.vn/invite?name=anhcodermayman?mode=duo và Anh Coder May Mắn tay run run bấm vào link không phải vì hồi hộp mà vì… mạng lag. Anh Coder May Mắn hiểu rằng: Nếu mình WA sub cuối thì tình bạn của hai người sẽ chấm dứt. Trước mắt họ là 1 loạt trận đấu xếp hạng – nếu thắng tất cả…  
họ sẽ chạm tới đỉnh cao mà chỉ truyền thuyết mới nhắc đến:

🏆 **Thách Đấu Người Ra Đề TBCP.**”

Bài 11: Tình yêu không đơn giản là những con số 0 và 1.- NOTSIMPLELOVE (1.0s, 4G)

**“Tình yêu trong anh vẫn luôn thầm lặng  
Nhưng không có nghĩa không rộng lớn  
Chỉ là anh đôi khi khó nói nên lời  
Mong em hãy cảm nhận thôi”**

Lúc này, An và Anh Coder May Mắn đã thấm mệt. Nhưng họ còn một nhiệm vụ cuối: Xác định tình yêu tối đa của K người yêu cũ dành cho họ và xem ai có tình yêu tối đa lớn nhất.

Tình yêu tối đa của một người được tính dựa trên một “dãy tình yêu” S có độ dài L chỉ gồm các kí tự 0 và 1. “Tình yêu tối đa của một người là độ dài của đoạn con dài nhất thỏa mãn các điều kiện dưới đây (0-indexed). Nếu không thể thỏa mãn, mặc định tình yêu tối đa = 0.

- Độ dài đoạn con phải là số nguyên tố > P.

- Số lượng đoạn con *ch* trong đoạn con đó cũng phải là số nguyên tố.

- Hai kí tự ở vị trí M và N của đoạn con phải giống nhau.

- Đoạn con ở vị trí (x, y) phải có ít nhất A đoạn con đối xứng có độ dài lớn hơn 1 trong nó

Input: Dòng đầu: số K, thể hiện số người yêu cũ

Mỗi testcase có cách thức như sau:

Dòng đầu: Tám biến L, P, *ch*, M, N, X, Y, A

Dòng sau: Dãy tình yêu S.

Output: Lần lượt in ra K dòng, dòng thứ i là tình yêu tối đa của người yêu cũ thứ i. Dòng cuối cùng, in ra chỉ số tình yêu tối đa lớn nhất và nhỏ nhất.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  1 1 1 1 1 1 1 1  0 | 0 |

Subtasks:

Subtask 1 (3%) : P < , L < , K < 100

Subtask 2 (15%) : < P <, < L < , 100 < K <

Subtask 3 (30%) ; < P < , < L < , < K < ,

Subtask 4 (52%) : < P < , < L < , < K <

“**Anh lặng lẽ xoá dòng cuối trong input.  
Dòng đó là ký ức cuối về em.**

**Chương trình trả về: "No Exact Cover".  
Anh biết. Đó là lúc phải bước sang Day 3”**