

SUMMER CONTEST NGÀY 2

Lưu ý: Các thí sinh có thể nộp bằng ngôn ngữ Pascal, C++ hoặc Java tùy theo ngôn ngữ nào thí sinh thấy thuận tiện. Tên các bài là Tenbai.*, trong đó * tương ứng với pas, cpp hay java. Tenbai là tên của bài sẽ được quy định trong cụ thể từng bài.

Bài 4 (41 điểm – SC4.*):

Aubameyang và Benzema, cùng chơi một trò chơi ăn hạt dẻ. Chúng ta gồm 3 cái tô, gồm 1 tô lớn đựng N hạt dẻ và 2 tô nhỏ cho mỗi người. Luật chơi sẽ diễn ra như sau:

- Người hiện tại sẽ bốc một lượng hạt tùy ý từ tô lớn, không được phép bốc 0 hạt và số hạt bốc được không vượt quá số M. Nếu số lượng hạt trong tô lớn nhỏ hơn hoặc bằng M, người hiện tại được quyền lấy hết. Người chơi hiện tại sẽ đặt những hạt mà mình bốc được vào tô nhỏ hiện tại của mình. Nếu trong tô lớn còn hạt, lượt chơi tiếp theo sẽ là của người đối thủ.
- Nếu tô lớn đã hết hạt dẻ, thì người vừa bốc được hạt dẻ cuối cùng sẽ ăn toàn bộ hạt dẻ mình đã bốc được. Đồng thời người còn lại phải bỏ toàn bộ hạt dẻ trong tô của mình vào lại tô lớn và lượt bốc bắt đầu từ người vừa mới đổ vào. Trò chơi kết thúc khi toàn bộ hạt dẻ đã ở trong bụng của 2 người chơi.

Bạn hãy xác định xem Aubameyang sẽ ăn được tối đa bao nhiêu hạt dẻ nếu Aubameyang là người chơi đầu tiên ?

- **Dữ liệu đầu vào (SC4.INP):**
 - o Đọc từ file SC4.INP
 - o Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 30$) là số lượng bộ dữ liệu của đề bài.
 - o T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức gồm 2 số nguyên dương N và M cách nhau bởi 1 dấu cách.
- **Dữ liệu đầu ra (SC4.OUT):**
 - o Ghi ra file SC4.OUT
 - o Với bộ dữ liệu thứ i, ta in ra 1 số nguyên dương duy nhất là số lượng hạt dẻ mà Aubameyang có thể ăn được tối đa nếu như Aubameyang là người thực hiện lượt chơi đầu tiên. Lưu ý là ta xem như cả 2 người đều vô cùng thông minh và đều chơi theo phương án sao cho mình ăn được nhiều nhất.
- **Ví dụ:**

SC4.INP	SC4.OUT
7	1
1 1	1

4 1	2
4 2	3
5 2	7
13 7	28
50 14	23
48 23	

- **Giải thích test ví dụ:**

- Với ví dụ 1: Chỉ có 1 hạt dẻ và tối đa bốc 1 hạt. Do đó Aubameyang bốc 1 hạt vào tô của mình. Vì đã hết hạt dẻ nên Aubameyang ăn 1 hạt mình vừa bốc được. Trò chơi kết thúc nên Aubameyang ăn được 1 hạt.
- Với ví dụ 2, lượt chơi như sau:
 - Aubameyang bốc 1, được 1
 - Benzema bốc 1, được 1
 - Aubameyang bốc 1, được 2
 - Benzema bốc 1, được 2
 - Lúc này tô lớn đã hết hạt dẻ, Benzema sẽ ăn 2 hạt của mình.
 - Aubameyang đổ 2 hạt của mình vừa bốc được vào lại tô lớn và lượt chơi bắt đầu từ Aubameyang.
 - Aubameyang bốc 1, được 1
 - Benzema bốc 1, được 1
 - Lúc này tô lớn hết hạt dẻ, Benzema ăn 1 hạt của mình. Aubameyang đổ 1 hạt mình bốc được vào tô lớn và lượt chơi bắt đầu lại từ Aubameyang.
 - Aubameyang bốc 1. Lúc này tô lớn đã hết hạt nên Aubameyang ăn 1 hạt mình bốc được. Như vậy Aubameyang chỉ có thể ăn được tối đa 1 hạt mà thôi :’(.
- Với ví dụ 3, lượt chơi diễn ra như sau:
 - Aubameyang bốc 2 hạt, được 2
 - Benzema bốc 2 hạt, được 2. Lúc này tô lớn đã hết hạt dẻ nên Benzema ăn 2 hạt, Aubameyang đặt 2 hạt dẻ của mình vào trong tô lớn.
 - Aubameyang bốc 2 hạt. Do tô lớn lúc này đã hết hạt dẻ nên Aubameyang ăn 2 hạt dẻ của mình. Tổng cộng Aubameyang ăn tối đa 2 hạt.
- Với ví dụ 4, lượt chơi như sau:
 - Aubameyang bốc 2, được 2
 - Benzema bốc 2, được 2
 - Aubameyang bốc 1, được 3
 - Lúc này tô lớn đã hết hạt dẻ. Aubameyang ăn 3 hạt mình đã bốc được và Benzema phải đặt lại 2 hạt mình đã bốc được vào tô lớn và lượt chơi bắt đầu từ Benzema.

- Benzema bốc 2 hạt, được 2
- Lúc này tô lớn đã hết hạt dẻ, nên Benzema ăn 2 hạt này. Trò chơi kết thúc và Aubameyang ăn được tối đa 3 hạt.
- Với ví dụ 6 và 7, là dùng để bạn đọc có thể tự kiểm chứng lại lời giải cho cả bài có đúng hay không.
- **Bài toán gồm 2 Dataset:**
 - Dataset 1 (11 điểm): $\frac{N}{2} \leq M \leq 50$.
 - Dataset 2 (30 điểm): $1 \leq M < \frac{N}{2}$.
 - Trong mọi Dataset, $1 \leq N \leq 50$.
- **Giới hạn thời gian và bộ nhớ:**
 - 5s / Dataset.
 - 512Mb / Dataset.

Bài 5 (28 điểm – SC5.*):

Bạn có một sàn nhà được tổ chức như một lưới hình chữ nhật kích thước $N * M$ gồm N dòng, M cột. Các ô của hình chữ nhật được đánh số theo thứ tự từ 1 đến $N * M$ theo chiều từ trái sang phải và từ trên xuống dưới.

Bạn thuê một chú rô bốt dùng để lau sàn nhà. Chú rô bốt này sẽ xuất phát từ góc trái trên của sàn nhà và lau dọn theo hình Zig Zag. Ví dụ giả sử với sàn nhà gồm 5 dòng và 6 cột, chú Rô bốt sẽ lau dọn theo thứ tự như sau:

1	2	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30

Chú rô bốt sẽ được lập trình để lau nhà như sau:

- Đầu tiên chú sẽ lau A ô theo đúng lộ trình như trên. Khi lau đến ô cuối, chú sẽ ghi nhớ tọa độ của ô này vào bộ nhớ.
- Sau đó chú tiếp tục lau thêm B ô nữa, tiếp tục lau sau khi đã lau A ô chứ không trở lại nơi xuất phát.
- Chú rô bốt sau khi lau xong A + B ô, sẽ trở lại tọa độ ô ban nãy mà chú đã ghi nhớ theo con đường nhanh nhất có thể. Biết là chú sẽ chạy theo con đường song song với các cạnh hệ trục tọa độ.

Bạn hãy giúp chú rô bốt tính xem khoảng cách từ ô thứ A + B đến tọa độ ô mà chú rô bốt nhớ là bao nhiêu. Khoảng cách được tính bằng khoảng cách Manhattan. Tài liệu về khoảng cách Manhattan: https://en.wiktionary.org/wiki/Manhattan_distance

- **Dữ liệu đầu vào (SC5.INP):**
 - o Đọc từ file SC5.INP.
 - o Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 20$) là số lượng bộ dữ liệu cần xử lý
 - o T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức như sau:
 - Gồm một dòng chứa 4 số nguyên dương duy nhất cách nhau bởi 1 khoảng cách, lần lượt là N, M, A, B.
- **Dữ liệu đầu ra (SC5.OUT):**
 - o Ghi vào file SC5.OUT.
 - o Gồm T dòng, dòng thứ i là kết quả của bộ dữ liệu thứ i, là độ dài khoảng cách Manhattan ta tính được.
- **Ví dụ:**

SC5.INP	SC5.OUT
6	2
5 6 2 2	3
5 6 9 5	0
5 6 9 0	450
500 1 30 450	718
829 910 1000 924	28160580732616308
287624057338675345 442885325636476446	
284187065224529461 28160580732616308	

- **Giải thích test ví dụ:**
 - o Với ví dụ 1, rô bốt sẽ lau nhà như sau:

<u>1</u>	<u>2</u>	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30

Ta thấy sau khi rô bốt lau 2 ô đầu, rô bốt lưu ô (1, 2) vào bộ nhớ. Sau đó lau tiếp 2 ô tiếp theo nữa, rô bốt sẽ đứng tại ô (1, 4). Từ ô (1, 4) để trở về ô (1, 2), rô bốt cần di chuyển 2 bước, tức khoảng cách Manhattan là 2.

- Với ví dụ 2:

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>
12	11	10	<u>9</u>	<u>8</u>	<u>7</u>
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30

Sau khi lau 9 ô đầu, rô bốt sẽ đứng ở vị trí (2, 4). Tiếp theo rô bốt lau 5 ô nữa, rô bốt sẽ đứng tại vị trí (3, 2). Khoảng cách Manhattan từ ô (2, 4) đến ô (3, 2) là 3.

- Với test ví dụ số 3, do $B = 0$ nên sau khi lau A ô, rô bốt sẽ đứng yên tại chỗ nên khoảng cách Manhattan là 0.

- 3 test ví dụ còn lại là dùng để cho các bạn tự kiểm tra xem thuật toán của mình đã đúng hay chưa.
- Test ví dụ cuối cùng sẽ không xuất hiện trong Small Dataset.
- **Bài toán sẽ được chia thành 2 Dataset:**
 - Small Dataset (12 điểm) : $1 \leq N, M \leq 2000$. $0 \leq A, B \leq N * M$.
 - Large Dataset (16 điểm): $1 \leq N, M \leq 10^{18}$. $0 \leq A, B \leq 10^{18}$.
 - Trong mọi bộ test, $A + B \leq N * M$.
- **Giới hạn dữ liệu:**
 - 5s / Dataset.
 - 512Mb / Dataset.

Bài 6 (31 điểm – SC6.*):

Bạn đang ở trong mê cung được tổ chức bởi N căn phòng và những con đường một chiều nối những căn phòng với nhau. Những căn phòng được đánh số từ 1 đến N. Ta có thể xem như đây là một đồ thị có hướng gồm N đỉnh và M cạnh 1 chiều. Để vào được phòng, ta cần phải có một mã số để có thể vào được căn phòng đó. Để mở được căn phòng thứ i, bạn phải có một con số thuộc đoạn $[l_i, r_i]$. Để mở cửa đi vào thì ta cần nhập mật khẩu, nhưng để đi ra thì không cần.

Bạn có K người công nhân, người thứ j sẽ có cho mình một mã số là j. Ban đầu cả K người đều đứng tại căn phòng S và mọi người đều muốn đến căn phòng E. Tất cả bọn họ trong quá trình di chuyển sẽ không thay đổi được mã số gắn riêng cho mình. Bạn hãy đếm xem có bao nhiêu người có thể đến được căn phòng E nhé.

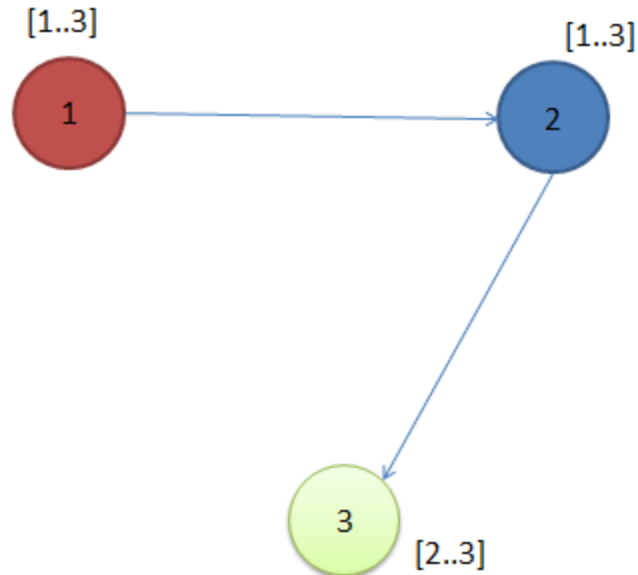
- **Dữ liệu đầu vào (SC6.INP):**
 - Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 20$), là số lượng bộ dữ liệu.
 - T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức dưới dạng như sau:
 - Dòng 1 gồm 5 số nguyên dương N, M, S, E, K ($1 \leq S, E \leq N$), tương ứng là số lượng căn phòng, số lượng đường một chiều nối giữa những căn phòng đó, chỉ số của căn phòng bắt đầu nơi của các công nhân đang ở, chỉ số của căn phòng nơi mà các công nhân muốn đến và số lượng người công nhân.
 - N dòng tiếp theo, dòng thứ i gồm 2 số l_i và r_i cách nhau bởi 1 dấu cách, tương ứng là chặn dưới và chặn trên của các mật khẩu để mà ta muốn vượt qua cửa thứ i.
 - M dòng tiếp theo, dòng thứ i gồm 2 số nguyên dương u_i và v_i ($1 \leq u_i, v_i \leq N$), thể hiện có 1 con đường 1 chiều nối từ căn phòng thứ u_i sang v_i .
- **Dữ liệu đầu ra (SC6.OUT):**
 - Ta in ra đáp án với T bộ dữ liệu tương ứng, bộ dữ liệu thứ i là số lượng người có thể đến được căn phòng E.

- Ví dụ:

SC6.INP	SC6.OUT
4	2
3 2 1 3 3	6
1 3	0
1 3	4
2 3	
1 2	
2 3	
4 5 1 4 10	
1 9	
1 4	
7 8	
3 8	
1 2	
1 3	
1 4	
2 3	
3 4	
4 4 1 4 10	
1 9	
1 4	
7 8	
3 8	
2 1	
3 1	
2 3	
3 4	
4 4 2 3 9	
1 4	
1 9	
3 8	
7 8	
1 3	
2 1	
2 4	
4 3	

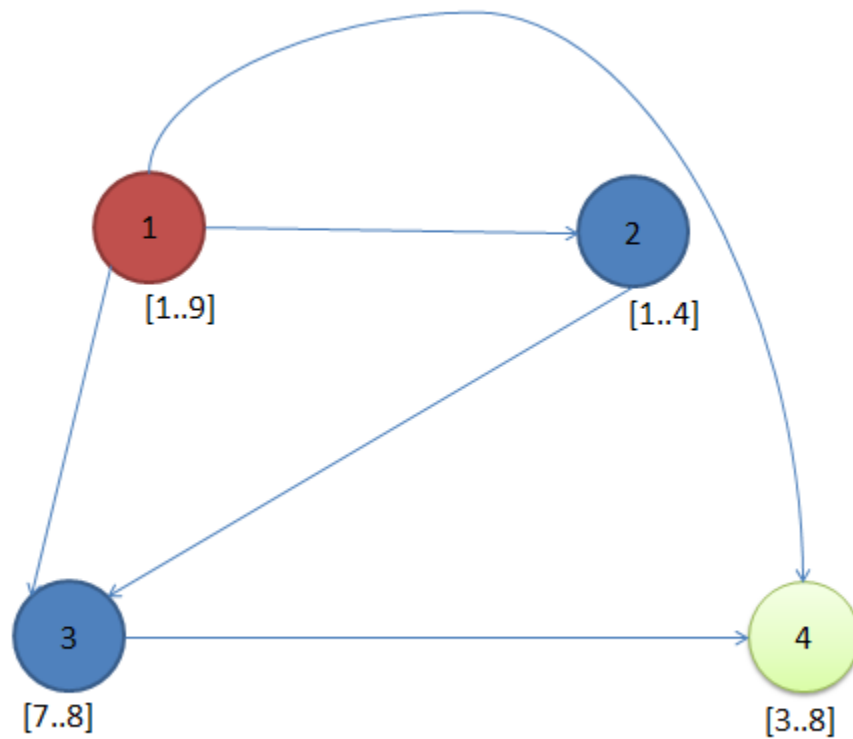
- Giải thích test ví dụ:

- Với test ví dụ 1:



Có tổng cộng 3 người công nhân, với 3 mã khóa là 1, 2 và 3. Ban đầu họ xuất phát tại đỉnh 1, ta thấy rằng để đến được đỉnh 2 thì cả 3 người họ đều vượt qua. Nhưng để đến được đỉnh 3, người 1 không thể nào vào được vì cửa 3 chỉ cho phép người có khóa 2 và 3 vào được thôi. Do đó đáp án là 2.

- Với test ví dụ 2:



Những người công nhân có khóa từ 3 đến 8 sẽ đi một mạch từ đỉnh 1 sang đỉnh 4. Như vậy có tổng cộng 6 người đến được đỉnh 4.

- Với 2 test ví dụ còn lại, bạn đọc có thể tự kiểm chứng bằng tay.

- **Bài toán sẽ được chia làm 2 Dataset:**

- Small Dataset (13 điểm): $1 \leq K \leq 2000$.
- Large Dataset (18 điểm): $1 \leq K \leq 10^9$.
- Với mọi Dataset, ta có $1 \leq N \leq 2000$, $1 \leq M \leq 5000$.

- **Giới hạn dữ liệu:**

- 5s / Dataset.
- 512Mb / Dataset.

-----Chúc các bạn làm bài tốt <3 <3 <3 -----