

Mục lục

Trò chơi Trăng Sao — KIDGAME	1
Máng rác – TRASH	3
BUSFREE	4

Bài 1. Trò chơi Trăng Sao

File dữ liệu vào: Gọi hàm
File kết quả: Trả về

Thành thấy một đám trẻ con đang chơi một trò chơi rất thú vị. Thành lại gần bắt chuyện:

- Chào các cháu, các cháu đang chơi trò gì thế?
- Chúng cháu có một trọng tài, còn những người còn lại được chia vào hai đội: đội Sao và đội Trăng. Có n cháu trong đội Sao và m cháu trong đội Trăng. Chú hãy đoán xem ai trong số chúng cháu là trọng tài rồi chúng cháu sẽ nói cho chú biết chi tiết trò chơi?"
- Đồng ý, thế luật chơi thế nào?
- Chú có thể hỏi ai đó nằm trong đội nào nhưng không được phép hỏi trực tiếp ai đó có phải là trọng tài hay không. Biết rằng người trong đội Sao sẽ luôn nói thật, còn người trong đội Trăng luôn nói dối, còn trọng tài thì trả lời tùy theo thứ tự câu hỏi: luôn nói dối ở những câu hỏi lẻ: nghĩa là những câu hỏi thứ nhất, thứ 3, thứ 5, ..., và luôn nói thật ở những câu hỏi chẵn: nghĩa là những câu hỏi thứ 2, thứ 4, thứ 6,
- Thế có hạn chế gì không?
- Chú không được hỏi một người về chính bản thân người đó. Ví dụ chú không được hỏi câu :“ Cháu ở đội Sao phải không?”. Thêm nữa chú chỉ được phép hỏi một cháu tối đa là hai câu hỏi và hai câu hỏi này phải hỏi về hai cháu khác nhau. Ví dụ chú hỏi Dũng liệu Tuấn có thuộc đội Sao không, thì sau đó chú không được hỏi Dũng là liệu Tuấn có thuộc đội Trăng không.
- Nhất trí. Vậy chú sẽ tìm ra tất cả các cháu, mỗi cháu thuộc đội nào, không chỉ tìm ra trọng tài thôi đâu.

Chương trình phải sử dụng một thư viện riêng `kidgamelib.h` (cho C++). Trong chương trình của bạn cần khai báo các thư viện này ở đầu chương trình:

```
#include "kidgamelib.h"
```

Thư viện cung cấp các hàm sau:

- Các hàm khởi tạo trò chơi
 - `int get_T();`
Chương trình phải gọi hàm này để khởi tạo trò chơi và chỉ được gọi duy nhất 1 lần. Hàm này trả về một giá trị T là số lượng bộ test.
 - `void get_nm(&n, &m);`
Chương trình phải gọi hàm này T lần sau hàm `get_T()`, lần thứ i hàm này trả về hai số n và m tương ứng là số cháu trong đội Sao và Trăng tương ứng với bộ test thứ i . Các cháu được đánh số từ 1 đến $n + m + 1$.

- Hàm thực hiện truy vấn:
`int ask(int a, int b, int d);`
Hỏi cháu a liệu cháu b có nằm trong đội d không? Với $1 \leq a, b \leq n + m + 1$, và $a \neq b$. $d = 0$ chỉ đội Sao và $d = 1$ chỉ đội Trăng. Hàm trả về 1 nếu câu trả lời là ‘Đúng’, còn trả về 0 nếu câu trả lời là ‘Sai’. Chỉ được phép hỏi một cháu về cùng một cháu khác tối đa 1 lần.
- Hàm trả lời câu hỏi:
`void answer(int *C);`
Các giá trị từ phần tử $C[1]$ đến phần tử thứ $C[n+m+1]$ chỉ vai trò từng cháu trong trò chơi: 0 nghĩa là thuộc đội Sao, 1 nghĩa là thuộc đội Trăng và 2 nghĩa là trọng tài. Phần tử $C[0]$ không quan trọng, nhận giá trị tùy ý.

Ví dụ

Gọi hàm	Trả về
<code>get_T()</code>	1
<code>get_nm(n,m)</code>	1 1
<code>ask(1,2,0)</code>	0
<code>ask(2,1,0)</code>	1
<code>ask(3,1,1)</code>	0
<code>answer(C)</code> với $C=[-1,2,1,0]$	Chương trình kết thúc

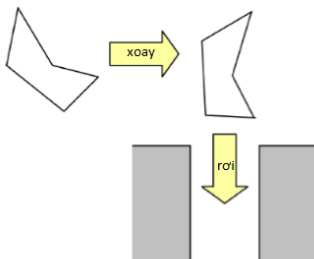
Hạn chế

$$2 \leq n + m \leq 500$$

Bài 2. Máng rác

ACM là một công ty xây dựng máng rác. Máng rác là một ống hình trụ được đặt trong các tòa nhà cao tầng. Rác được bỏ vào máng sẽ rơi xuống dưới và được thu gom tại tầng hầm. Máng phải được thiết kế vừa với các loại rác được thả vào. Do kích thước của máng càng lớn thì càng tốn chi phí, nên các công ty luôn muốn xây dựng máng rác nhỏ nhất có thể.

Để đơn giản hóa vấn đề, chúng ta sẽ chỉ xem xét việc thiết kế máng rác dưới dạng 2D. Một máng rác dạng thẳng đứng và chiều rộng cố định. Các đồ vật được thả vào máng có là các đa giác. Trước khi thả, các đa giác có thể được xoay để vừa với máng, và chúng sẽ không bị xoay trong quá trình rơi.



Nhiệm vụ: tính toán độ rộng bé nhất của máng rác để một đa giác cho trước lọt qua.

Dữ liệu vào

Đầu vào bao gồm nhiều test (số lượng test < 400). Mỗi test bắt đầu bằng một số nguyên n ($3 \leq n \leq 100$) chỉ số đỉnh của đa giác. Tiếp theo là n dòng chứa các cặp x_i và y_i ($0 \leq x_i, y_i \leq 10^4$) cho biết tọa độ các đỉnh của đa giác. Tất cả các điểm này đôi một khác nhau và đảm bảo các cạnh đa giác không cắt nhau (trừ tại các đỉnh của đa giác).

Test cuối cùng theo sau bởi một số 0.

Kết quả

Kết quả mỗi test tương ứng ghi trên một dòng chỉ số test (Case i:) tiếp theo là một giá trị là độ rộng nhỏ nhất của máng rác với độ chính xác đến hai chữ số thập phân sau dấu phẩy.

Ví dụ

test	answer
3 0 0 3 0 0 4 4 0 10 10 0 20 10 10 20 0	Case 1: 2.40 Case 2: 14.14

Bài 3. BUSFREE

File dữ liệu vào: `input_xy.txt`
File kết quả: `output_xy.txt`

Một bản đồ xe buýt được cho trong một hình chữ nhật kích thước $p \times q$ trên hệ tọa độ Oxy, tọa độ điểm dưới trái là $(0, 0)$, điểm trên phải là (p, q) . Có n tuyến xe buýt, mỗi tuyến gồm hai chặng tương ứng với hai đường gấp khúc có chung hai điểm đầu cuối, nằm trọn vẹn trong bản đồ. Có c ($c \leq p \times q$) ô vuông quan trọng nằm trong bản đồ, mỗi ô vuông có cạnh dài 1 đơn vị, các cạnh song song với trục tọa độ và các đỉnh có tọa độ nguyên. Mỗi tuyến xe chỉ có đúng 1 xe buýt duy nhất chạy.

Có m ($m \leq n$) sensor cần được đặt lên các xe buýt để quan sát các điểm quan trọng. Mỗi khi được bật lên, sensor sẽ quan sát được tất cả các ô vuông quan trọng có điểm chung với hình tròn bán kính r tính từ vị trí bật sensor. Vì hạn chế năng lượng, mỗi sensor sẽ chỉ được phép bật không quá c lần trên mỗi chặng và sẽ được tắt ngay sau khi quan sát xong (thời gian quan sát rất nhỏ, có thể coi như bằng 0).

Yêu cầu: Chọn ra m tuyến xe để đặt sensor và các vị trí bật sensor trên mỗi tuyến nhằm tối đa hóa số lượng ô vuông quan trọng có thể quan sát được.

Đây là bài toán chỉ cần nộp các file kết quả đầu ra. Thí sinh được cho 10 file đầu vào tương ứng với 10 test, đối với mỗi file đầu vào thí sinh cần nộp một file kết quả đầu ra tương ứng. Với mỗi file kết quả đầu ra, điểm của thí sinh phụ thuộc vào độ tốt của đáp án so với đáp án của ban tổ chức (xem cách tính điểm trong phần Chấm điểm).

Dữ liệu vào

Thí sinh được cung cấp 10 file dữ liệu đầu vào với tên tương ứng là: `input_00.txt`, `input_02.txt`, ..., `input_09.txt`. Mỗi file dữ liệu đầu vào có khuôn dạng như sau:

- Dòng đầu ghi hai số nguyên dương p, q và số thực không âm r .
- Dòng thứ hai ghi hai số nguyên dương n, m .
- Các dòng tiếp theo ghi thông tin của từng tuyến xe buýt. Mỗi tuyến xe buýt gồm có:
 - Dòng đầu ghi tổng số điểm gấp khúc trên toàn tuyến.
 - Dòng tiếp theo ghi số điểm gấp khúc của chặng thứ nhất và theo sau đó là tọa độ các điểm gấp khúc của chặng thứ nhất, mỗi tọa độ nằm trên một dòng.
 - Dòng tiếp theo đó là số điểm gấp khúc của chặng thứ hai và theo sau là tọa độ các điểm gấp khúc của chặng thứ hai, mỗi tọa độ nằm trên một dòng.
- Dòng tiếp theo ghi số nguyên không âm c .
- c dòng tiếp theo ghi tọa độ góc trái dưới của các ô vuông quan trọng.

Kết quả

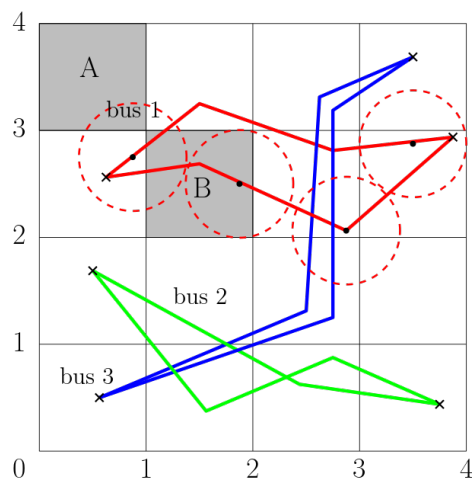
Đối với mỗi file dữ liệu đầu vào, thí sinh cần nộp một file kết quả đầu ra có tên tương ứng là: `output_00.txt`, `output_02.txt`, ..., `output_09.txt`. Mỗi file kết quả đầu ra có khuôn dạng:

- Dòng đầu in ra số ô vuông quan trọng có thể quan sát được.
- Các dòng tiếp theo in ra các tuyến xe buýt được lắp sensor và các vị trí mà sensor được bật trên tuyến đó. Mỗi tuyến gồm có:
 - Dòng đầu tiên in ra số hiệu của tuyến xe.
 - c dòng tiếp theo, mỗi dòng gồm tất cả các ô vuông quan trọng mà một vị trí bật sensor trên chặng thứ nhất có thể quan sát được. Mỗi dòng sẽ bao gồm một số nguyên không âm d là số ô vuông quan trọng quan sát được, theo sau là tọa độ của d ô vuông quan trọng đó.
 - c dòng sau đó, mỗi dòng gồm tất cả các ô vuông quan trọng mà một vị trí bật sensor trên chặng thứ hai có thể quan sát được. Mỗi dòng sẽ bao gồm một số nguyên không âm d là số ô vuông quan trọng quan sát được, theo sau là tọa độ của d ô vuông quan trọng đó.

Ví dụ

input_xy.txt	output_xy.txt
4 4 0.5	2
3 1	1
8	2 0 3 1 2
4	0
0.625 2.5625	0
1.5 3.25	1 1 2
2.75 2.8125	
3.875 2.9375	
4	
3.875 2.9375	
2.875 2.0625	
1.5 2.6875	
0.625 2.5625	
7	
3	
0.5 1.6875	
2.3475 0.625	
3.75 0.4375	
4	
3.75 0.4375	
2.75 0.875	
1.6875 0.375	
0.5 1.6875	
8	
4	
0.5625 0.5	
2.5 1.3125	
2.625 3.3125	
3.5 3.6875	
4	
3.5 3.6875	
2.75 3.1875	
2.75 1.25	
0.5625 0.5	
2	
0 3	
1 2	

Giải thích



Có 3 tuyến xe và 2 ô quan trọng là A và B. Đáp án mẫu chọn tuyến xe 1 để lắp đặt sensor. Với các vị trí bật

sensor như trên hình, cả hai ô quan trọng đều được quan sát. Trên chặng thứ nhất, vị trí bật sensor thứ nhất sẽ quan sát được cả A và B, vị trí bật thứ hai không theo dõi được ô nào. Trên chặng thứ hai, vị trí bật sensor thứ nhất không theo dõi được ô nào, trong khi vị trí bật thứ hai sẽ theo dõi được ô vuông B.

Hạn chế

- $1 \leq p, q \leq 50$
- $n = 60$ trong tất cả các test

Chấm điểm

Với mỗi file dữ liệu đầu vào, phần trăm điểm mà thí sinh nhận được tính bằng công thức: $100 \times \min(1, \frac{S_{ts}^2}{S_{btc}^2})(\%)$. Trong đó S_{btc} và S_{ts} tương ứng là số ô vuông quan trọng quan sát được trong đáp án của ban tổ chức và đáp án trong file kết quả đầu ra của thí sinh.

Thí sinh có thể nộp từng file kết quả đầu ra tương ứng với file dữ liệu đầu vào hoặc nén các file kết quả đầu ra thành một file có tên submission.zip để nộp. Điểm số của mỗi test là điểm cao nhất đạt được trong các lần nộp file kết quả đầu ra của test đó. Điểm số của bài là tổng điểm của từng test.

Lưu ý

Một số công thức hữu ích:

- Phương trình đường thẳng đi qua hai điểm (x_1, y_1) và (x_2, y_2) là:

$$(y_2 - y_1)x - (x_2 - x_1)y - (y_2 - y_1)x_1 + (x_2 - x_1)y_1 = 0$$

- Công thức tìm giao điểm của đường tròn tâm $(0, 0)$ bán kính r và đường thẳng $ax + by + c = 0$:
 - Nếu $c^2 > r^2(a^2 + b^2)$ thì hai đường không cắt nhau.
 - Nếu $c^2 = r^2(a^2 + b^2)$ thì hai đường có giao điểm duy nhất $(\frac{-ac}{a^2+b^2}, \frac{-bc}{a^2+b^2})$.
 - Nếu $c^2 < r^2(a^2 + b^2)$ thì hai đường có hai giao điểm.
- + Đặt $d = r^2 - \frac{c^2}{a^2+b^2}$
- + Đặt $mul = \sqrt{\frac{d}{a^2+b^2}}$
- + Hai giao điểm là: $(\frac{-ac}{a^2+b^2} + b * mul, \frac{-bc}{a^2+b^2} - a * mul), (\frac{-ac}{a^2+b^2} - b * mul, \frac{-bc}{a^2+b^2} + a * mul)$