



OLYMPIC TIN HỌC SINH VIÊN LẦN THỨ XXV, 2016

Khối thi: Siêu cúp

Thời gian làm bài: 180 phút

Ngày thi: 30-11-2016

Nơi thi: ĐẠI HỌC NHA TRANG

TỔNG QUAN ĐỀ THI

Tên bài	Tên file chương trình	Tên file dữ liệu	Tên file kết quả
Dàn đèn màu	LAMPS.???	LAMPS.INP	LAMPS.OUT
Đổi bánh	CAKE.???	CAKE.INP	CAKE.OUT
Hệ thống rada	RADA.???	RADA.INP	RADA.OUT
Alice ở xứ sở 01	ALICE.???	Không có	Không có

Chú ý: Dấu ??? được thay thế bởi đuôi ngầm định của ngôn ngữ được sử dụng để cài đặt chương trình.

Hãy lập trình giải các bài sau đây:

Bài 1. Dàn đèn màu

Bờm nhận nhiệm vụ lắp đặt một hệ thống dàn đèn màu trang trí cho cổng vào khu biệt thự nghỉ dưỡng “Cây xương rồng” của Phú ông. Dàn đèn mà Bờm lắp đặt gồm các ống đèn nhiều màu sắc được lắp nối tiếp với nhau. Phú ông khi đến nghiệm thu lại nảy ra một ý tưởng mới và yêu cầu Bờm cần gỡ bớt một số ống đèn màu để đáp ứng thị hiếu của mình.

Một cách chính xác, vấn đề mà Bờm cần giải quyết có thể mô tả như sau: Cho đơn đồ thị vô hướng liên thông $G = (V, E)$ với tập đỉnh $V = \{1, 2, \dots, n\}$ và tập cạnh E , trong đó mỗi cạnh $e \in E$ thuộc vào không quá một chu trình đơn. Nhắc lại là, trong chu trình đơn không có đỉnh nào bị lặp lại, ngoại trừ đỉnh đầu trùng với đỉnh cuối. Mỗi cạnh của đồ thị (tương ứng với một ống đèn màu trong dàn đèn) được gán với một màu nào đó. Chỉ số màu của các cạnh là các số nguyên dương. Cần tìm cách loại bỏ khỏi đồ thị G một số ít nhất các cạnh sao cho đồ thị thu được là đồ thị liên thông không chứa chu trình và đồng thời tổng số lượng màu khác nhau của các cạnh còn lại là lớn nhất.

Yêu cầu: Hãy giúp Bờm tính số lượng màu của các cạnh của đồ thị thu được thỏa mãn yêu cầu nêu trên.

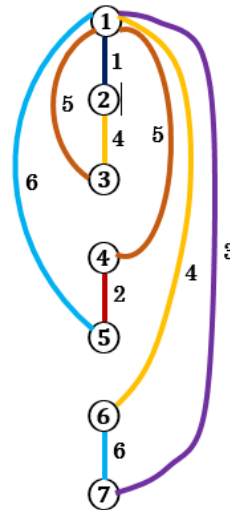
Dữ liệu: Vào từ file văn bản LAMPS.INP:

- Dòng đầu tiên chứa hai số nguyên dương n, m ($2 \leq n \leq 10\,000$) theo thứ tự là số lượng đỉnh và số lượng cạnh của đồ thị;
- Mỗi dòng trong số m dòng tiếp theo chứa ba số nguyên u, v, c ($1 \leq u, v \leq n, u \neq v, 1 \leq c \leq m$) cho biết trong đồ thị có cạnh nối 2 đỉnh u và v và màu của cạnh này là c . Dữ liệu đảm bảo đồ thị được mô tả là đơn đồ thị liên thông và mỗi cạnh của nó thuộc vào không quá một chu trình đơn. Hai số liên tiếp trên cùng dòng được ghi cách nhau ít nhất một dấu cách.

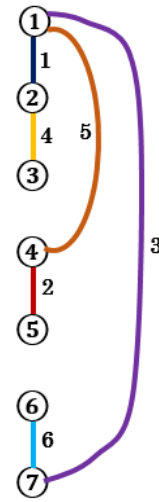
Kết quả: Ghi ra file văn bản LAMPS.OUT một số là số lượng màu tìm được.

Ví dụ:

LAMPS . INP	LAMPS . OUT
7 9 1 2 1 2 3 4 3 1 5 1 4 5 4 5 2 5 1 6 1 6 4 6 7 6 7 1 3	6



Hình 1a



Hình 1b

Giải thích: Đồ thị mô tả bởi ví dụ được cho trong Hình 1a. Ta cần loại bỏ các cạnh (1,5), (1,3) và (1,6) để thu được đồ thị trong Hình 1b có số lượng màu khác nhau trên các cạnh là 6.

Bài 2. Đổi bánh

Sau khi trúng giải độc đắc của xổ số điện toán, ông Long tổ chức một bữa tiệc lớn mời người thân cùng bà con láng giềng gần xa đến dự. Có tất cả n người (được đánh số từ 1 đến n) tới dự bữa tiệc. Kết thúc bữa tiệc, ông Long đã chuẩn bị m chiếc bánh để làm quà. Chiếc bánh thứ i có khối lượng w_i và được chuẩn bị dành cho người p_i . Đồng thời, ông Long đã sắp xếp chu đáo để đảm bảo tất cả mọi người tới dự tiệc đều có bánh mang về. Khi ra về, mọi người nói chuyện vui vẻ với nhau và hai người nào đó có thể đổi bánh cho nhau, mỗi người đưa người kia một chiếc bánh trong phần quà của mình. Một cách đổi bánh giữa hai người được gọi là thân thiện nếu như chênh lệch giữa tổng khối lượng bánh đang có của hai người là giảm đi sau khi thực hiện trao đổi bánh. Là một người rất mê điện toán, ông Long muốn đếm xem có bao nhiêu cách đổi bánh thân thiện có thể thực hiện giữa các vị khách của mình. Hai cách đổi bánh được gọi là khác nhau nếu tồn tại ít nhất một chiếc bánh xuất hiện trong cách đổi này nhưng không xuất hiện trong cách đổi kia.

Yêu cầu: Bạn hãy giúp ông Long giải quyết vấn đề đặt ra.

Dữ liệu: Vào từ file văn bản CAKE.INP với dòng đầu tiên chứa số nguyên dương T ($T \leq 10$) là số lượng test. Tiếp đến là T nhóm dòng, mỗi nhóm cho thông tin về một test theo khuôn dạng sau:

- Dòng đầu tiên chứa 2 số nguyên dương n, m ($n \leq m \leq 2 \times 10^5$);
 - Dòng thứ 2 chứa m số nguyên dương w_1, w_2, \dots, w_m ($w_i \leq 10^9$);
 - Dòng thứ 3 chứa m số nguyên dương p_1, p_2, \dots, p_m ($p_i \leq n$).
- Hai số liên tiếp trên cùng dòng được ghi cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra file văn bản CAKE.OUT T dòng, mỗi dòng ghi một số nguyên duy nhất là số lượng cách đổi bánh thân thiện tìm được đối với bộ dữ liệu vào tương ứng.

Ví dụ:

CAKE . INP	CAKE . OUT
1	4
4 5	
10 3 6 8 9	
2 3 2 1 4	

Giải thích: Người thứ nhất nhận bánh số 4 có khối lượng 8, người thứ hai nhận hai bánh số 1 và 3 với khối lượng tương ứng là 6 và 10, người thứ ba nhận bánh số 2 có khối lượng 3, người thứ tư nhận bánh số 5 có khối lượng 9.

Có 4 cách đổi bánh thân thiện:

- Người 1 đưa người 2 bánh khối lượng 8, người 2 đưa người 1 bánh khối lượng 10.
- Người 2 đưa người 3 bánh khối lượng 6, người 3 đưa người 2 bánh khối lượng 3.
- Người 2 đưa người 3 bánh khối lượng 10, người 3 đưa người 2 bánh khối lượng 3.
- Người 2 đưa người 4 bánh khối lượng 10, người 4 đưa người 2 bánh khối lượng 9.

Bài 3. Hệ thống Ra-đa

Để phát hiện xâm nhập sớm tại các vùng thiết yếu trên biển Đông, Bộ Tư lệnh Hải quân đã đặt hệ thống gồm n ra-đa bắt tín hiệu trên vùng lãnh hải quốc gia. Ra-đa i đặt ở một vị trí có tọa độ là (x_i, y_i) và có thể theo dõi toàn bộ lãnh hải trong vùng có dạng hình tròn tâm (x_i, y_i) bán kính r_i (ta sẽ gọi tắt là vùng bao phủ của ra-da i). Hệ thống ra-đa được đặt sao cho các vùng thiết yếu được bao phủ bởi nhiều ra-đa khác nhau. Với mỗi điểm trên lãnh hải, Bộ Tư Lệnh dễ dàng xác định được tập các ra-đa có vùng bao phủ chứa nó. Bộ Tư Lệnh quan tâm đến điểm thuộc vùng bao phủ của nhiều ra-đa nhất.

Yêu cầu: Hãy giúp Bộ Tư lệnh xác định số lượng phần tử của tập gồm một số nhiều nhất các ra-đa mà có ít nhất một điểm thuộc vào tất cả các vùng bao phủ của chúng.

Dữ liệu: Vào từ file văn bản RADA.INP với dòng đầu tiên chứa duy nhất một số nguyên dương T ($T \leq 10$) là số lượng test. Tiếp theo là T nhóm dòng, mỗi nhóm tương ứng với một test theo khuôn dạng sau:

- Dòng thứ nhất chứa duy nhất một số nguyên dương n .
- Dòng thứ i trong số n dòng tiếp theo chứa 3 số nguyên x_i, y_i, r_i , mỗi số có trị tuyệt đối không vượt quá 10^6 , tương ứng là các tọa độ của vị trí đặt thiết bị của ra-đa thứ i và bán kính bao phủ của nó.

Hai số liên tiếp trên cùng dòng được ghi cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra file văn bản RADA.OUT T dòng, mỗi dòng một số nguyên là các câu trả lời tương ứng cho T bộ dữ liệu.

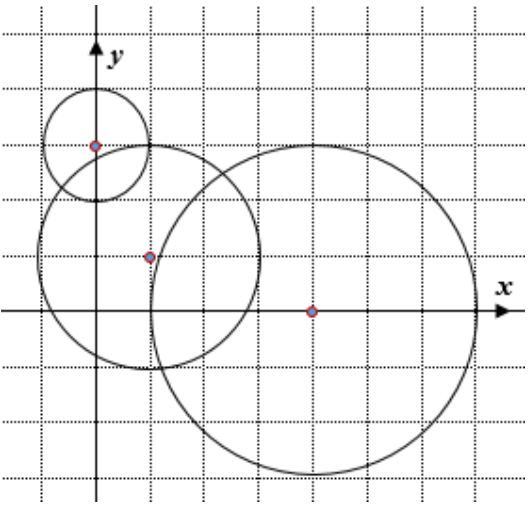
Subtask 1: Giả thiết là $n \leq 20$; x_i, y_i, r_i , mỗi số có trị tuyệt đối không vượt quá 10.

Subtask 2: Giả thiết là $n \leq 50$.

Subtask 3: Giả thiết là $n \leq 200$.

Subtask 4: Giả thiết là $n \leq 1000$.

Ví dụ:

RADA.INP	RADA.OUT	Hình vẽ minh họa
1 3 1 1 2 4 0 3 0 3 1	2	

Bài 4. Alice ở xứ sở 01

Mọi lần Alice lạc vào xứ sở thần tiên, nhưng lần này Alice lại lạc vào xứ sở của những con số 0 và 1. Do không hứng thú với xứ sở này nên Alice muốn nhanh chóng quay lại thế giới thực của mình. Alice đã tìm đến cửa ra thông minh nhưng để mở được cánh cửa phải giải mã được câu hỏi sau:

“Cho biết các số đứng ở một vài vị trí trên một dãy nhị phân độ dài n ($1 \leq n \leq 10^5$) (các vị trí trên dãy được đánh số từ 0 đến $n-1$), hỏi dãy có chứa hai số 0 và 1 (theo đúng thứ tự 0 rồi đến 1) ở hai vị trí liên tiếp nhau hay không?”.

Để có đủ dữ kiện trả lời câu hỏi này, cánh cửa thông minh cho phép Alice đặt ra các truy vấn, mỗi truy vấn có dạng sau:

“Số đứng ở vị trí thứ i của dãy là số nào?”.

Câu trả lời cho mỗi truy vấn như vậy chỉ là 0 hoặc 1. Cánh cửa thông minh chỉ mở khi Alice trả lời đúng cho câu hỏi đặt ra với số lần thực hiện truy vấn là ít nhất có thể.

Yêu cầu: Hãy giúp Alice viết chương trình tìm cách thoát khỏi xứ sở 01.

Chương trình phải sử dụng một thư viện riêng. Thư viện bao gồm các file sau: `alicelib.java` (cho java), `alicelib.h` và `alicelib.c` (cho C/C++).

Thư viện cung cấp các hàm sau:

- **Hàm khởi tạo trò chơi**

- **Đối với C/CPP:** `int get_n(int *a)`

- **Đối với Java:** `int get_n(List<Integer> a)`, trong đó mảng `a` là một biến kiểu

List<Integer> cần được khởi tạo trước khi truyền vào để gọi hàm.

Chương trình phải gọi hàm này để khởi tạo trò chơi. Hàm `get_n` trả về một giá trị n là độ dài của dãy nhị phân và mảng a độ dài n với các phần tử là 0 hoặc 1 hoặc -1, trong đó các vị trí đứng số 0 (hoặc số 1) là các vị trí mà số đứng ở đó được biết trước là 0 (hoặc 1), còn các vị trí đứng số -1 là các vị trí mà số đứng tại đó là chưa được biết.

- **Hàm thực hiện truy vấn:** `int get_01(int i)`. Hàm này trả về số đứng ở vị trí thứ i (là câu trả lời cho truy vấn về vị trí thứ i).
- **Hàm trả lời câu hỏi:** `void guess(int res)`. Để kết thúc chương trình cần gọi hàm này với `res` bằng 1 nếu câu trả lời cho câu hỏi đặt ra là khẳng định và bằng 0 nếu câu trả lời là phủ định. Sau khi gọi hàm này chương trình sẽ tự động kết thúc. Số lượng câu hỏi của chương trình của bạn sẽ bằng tổng số lần gọi hàm `get_01`.

Lưu ý: Mỗi hàm `get_n` và `guess` chỉ được gọi một lần duy nhất.

Nhiệm vụ của bạn là phải cài đặt hàm `run()` để gọi các hàm được cung cấp sẵn, xử lý và gọi hàm `guess` để đoán kết quả khi đã xác định được đáp án.

- Đối với C/C++ thì hàm này đã được khai báo header trong file `alicelib.h`. Bạn cần include file thư viện này và cài đặt hàm `run()` theo mẫu:

```
#include "alicelib.h"
void run() {
    // your code here
}
```

- Đối với Java thì cần khai báo class public tên `alice` và khai báo hàm `run()`. Cần chắc chắn hàm `run()` được để public:

```
public class alice{
    public void run(){
        // your code here
    }
}
```

Bạn có thể xem các file được cung cấp trên hệ thống để hiểu rõ hơn về cách tương tác với hệ thống.

Ví dụ:

Gọi hàm	Giá trị trả về
<code>get_n(a)</code>	5; $a = [-1, 1, 0, 0, -1]$; độ dài của dãy là 5, các vị trí 1, 2, 3 là các vị trí được biết trước
<code>get_01(4)</code>	0
<code>get_01(0)</code>	0
<code>guess(1)</code>	Kết thúc. Chương trình đã trả lời đúng với số lần thực hiện truy vấn là ít nhất. Cánh cửa thông minh sẽ tự động mở ra.

----- HẾT -----