Bài 1: Đi tìm kho báu

Ở xứ sở thần tiên nọ, có những khu đất bay lơ lửng trên trời. Giữa các khu đất có những cây cầu nối với nhau. Ở giữa mỗi cây cầu sẽ có một kho báu. Lưu ý rằng:

* Để lấy được kho báu thì bạn bắt buộc phải đi qua toàn bộ cây cầu.
* Sau khi đi qua thì cây cầu sẽ tự động biến mất.
* Nếu một khu đất bình thường khi không còn cây cầu nào nối với nó nữa thì nó cũng sẽ tự động biến mất. Ở xứ sở thần tiên này chỉ có tối đa hai khu đất bất thường.

Nhiệm vụ của bạn là hãy xác định đường đi (nếu có) bắt đầu và kết thúc đều ở những khu đất bất thường và có thể lấy toàn bộ kho báu trên đường đi.

Dữ liệu vào từ tập tin KHOBAU.IN: Dòng đầu tiên gồm hai số nguyên *N* và *M* trong đó *N* là số khu đất và *M* là số cây cầu thỏa mãn 0 ≤ *N, M* ≤ 100. Các khu đất sẽ được đánh số từ 1 tới *N*. *M* dòng tiếp theo mỗi dòng gồm hai số nguyên dương *u* và *v* cho biết có cầu nối giữa khu đất *u* và *v*. Dòng tiếp theo gồm số nguyên một số nguyên *K* tương ứng với số lượng khu đất bất thường và *K* số nguyên tiếp theo tương ứng với chỉ số của các khu đất bất thường.

Dữ liệu xuất ra tập tin tin KHOBAU.OUT: Nếu không tìm được đường đi như vậy, xuất ra ‘NO’. Ngược lại, dòng đầu xuất ra ‘YES’ và dòng tiếp theo xuất ra thứ tự đường đi.

|  |  |
| --- | --- |
| KHOBAU.IN | KHOBAU.OUT |
| 4 4  1 2  1 3  1 4  2 3  2 1 3 | YES  1 2 3 4 1 3 |
| 2 1  1 2  1 1 | NO |
| 3 2  1 2  2 3  2 1 3 | YES  1 2 3 |
| 3 2  1 2  2 3  0 | NO |

Bài 2: Dãy ngoặc

Cho một ký tự chỉ gồm dấu mở ngoặc tròn ‘(‘ và dấu đóng ngoặc tròn ‘)’. Ta có dãy ngoặc rỗng và dãy ngoặc ‘()’là hai dãy ngoặc hợp lệ. Ngòai ra, nếu K là dãy ngoặc hợp lệ thì ‘(K)’, ‘K()’ và ‘()K’ cũng là các dãy ngoặc hợp lệ. Cho trước một dãy ngoặc có độ dài không quá 1000. Hãy xuất ra hai số nguyên *a* và *b* tương ứng với số dấu mở ngoặc tròn và dấu đóng ngoặc tròn ít nhất cần thêm vào để dãy ngoặc cho trước trở thành dãy ngoặc hợp lệ.

Dữ liệu vào từ tập tin DAYNGOAC.IN: Chỉ gồm một dòng duy nhất mô tả chuỗi ngoặc.

Dữ liệu xuất ra tập tin DAYNGOAC.OUT: Xuất ra hai số *a* và *b*.

|  |  |
| --- | --- |
| DAYNGOAC.IN | DAYNGOAC.OUT |
| )()())((( | 2 3 |
| )))(( | 3 2 |
| )()(()( | 1 2 |
| )((()(()(( | 1 5 |
| )()(()(()( | 1 3 |

Bài 3: Chuỗi hình tròn

Cho *N* (*N ≤* 1000)hình tròn được mô tả dưới dạng *Oi*(*xi*, *yi*, *ri*) trong đó *xi*, *yi*, *ri* đều là các số nguyên thuộc khoảng [-10000, 10000] và *ri* ≠ *rj* với mọi *i* ≠ *j*. Hình tròn *A* được xem là có kết nối với hình tròn *B* nếu giữa *A* với *B* có phần chung và *rA* > *rB*. Chuỗi hình tròn *O1*, *O2*, …, *Om* gọi là chuỗi hình tròn có kết nối với nhau nếu như *Oj* có kết nối với *Oj+1* với *j* = 1, 2, …, *m*-1. Hãy cho biết chuỗi hình tròn có kết nối với nhau dài nhất và độ dài tương ứng *L* của chuỗi.

Dữ liệu vào từ tập tin CHUOIHINHTRON.IN: Dòng đầu tiền gồm duy nhất số nguyên *N*. *N* dòng tiếp theo mỗi dòng gồm ba số nguyên *xi*, *yi*, *ri* dùng để mô tả hình tròn *Oi.*

Dữ liệu xuất ra tập tin CHUOIHINHTRON.OUT: Dòng thứ nhất chứa duy nhất một số nguyên *L*. Dòng thứ hai chứa chỉ số của các hình tròn. Lưu ý, thứ tự xuất hiện của các hình tròn theo giảm dần của bán kính.

|  |  |
| --- | --- |
| CHUOIHINHTRON.IN | CHUOIHINHTRON.OUT |
| 0 | 0 |
| 5  0 0 1  2 2 2  200 200 1  -200 -200 1  200 -200 1 | 2  1 2 |
| 3  0 0 1  4 4 3  1 1 2 | 3  1 3 2 |

Bài 4: Gói hàng

Cho *N* món hang có trọng lượng khác nhau. Dùng cân dĩa để so sánh trọng lượng của các món hàng và ghi lại kết quả. Ví dụ như nếu món hàng thứ *u* nặng hơn món hang thứ *v* thì chúng ta sẽ ghi nhận trong bảng kết quả là *u* > *v*. Tuy nhiên, kết quả ghi nhận có thể có nhầm lẫn. Nghĩa là, tồn tại món hàng hàng *u*, *v* và *t* thỏa mãn *u* > *v*, v > *t* nhưng *t* > *u*. Hãy viết chương trình kiểm tra có sai sót gì trong kết quả ghi nhận hay không.

Dữ liệu vào vào từ tập tin GOIHANG.IN: Dòng đầu tiên gồm duy nhất số nguyên *N* và *M* trong đó *N* là số món hang và *M* là số lượng kết quả so sánh ghi nhận được (0 ≤ *N*, *M* ≤ 100). *M* dòng tiếp theo mỗi dòng gồm hai số nguyên dương *u* và *v* cho biết món hàng thứ *u* nặng hơn món hàng thứ *v*.

Dữ liệu xuất ra tập tin GOIHANG.OUT: Dòng đầu xuất ra 0 hoặc 1 với 0 nghĩa là kết quả ghi nhận không có lỗi và 1 nghĩa là có lỗi. Nếu có lỗi, dòng tiếp theo xuất ra hai số nguyên *u* và *v* cho biết có mẫu thuẫn giữa *u* và *v*. Nếu có nhiều trường hợp bị sai, xuất ra một trường hợp bất kỳ.

|  |  |
| --- | --- |
| GOIHANG.IN | GOIHANG.OUT |
| 4 4  1 > 2  1 > 3  1 > 4  2 > 3 | 0 |
| 2 1  1 > 2 | 0 |

Bài 5: Byteland

Ở vương quốc Byteland, nhu cầu đi lại bằng xe cộ của người dân rất cao nhưng xăng dầu rất đắt đỏ. Do đó, người dân cần biết số lít xăng tối thiểu để đi từ nút giao thông A đến nút giao thông B.

Dữ liệu vào từ tập tin BYTELAND.IN với định dạng như sau: Dòng đầu tiên chứa hai số nguyên N, M lần lượt tương ứng với số nút giao thông và số tuyến đường của vương quốc. M dòng tiếp theo mỗi dòng có ba số nguyên u, v, d cho biết để đi từ nút giao thông thứ u tới nút giao thông thứ v hết d lít xăng. Ngược lại, đi từ nút giao thông thứ v tới nút giao thông thứ t cũng hết d lít xăng. Dòng cuối cùng chứa hai số nguyên A và B thể hiện mong muốn của người dẫn muốn biết đi từ nút giao thông A tới nút giao thông B hết bao nhiêu lít xăng.

Dữ liệu xuất sẽ được ghi ra tập tin BYTELAND.OUT với nội dung như sau: Gồm một số nguyên d duy nhất tương ứng với số lít xăng tối thiểu.

Lưu ý: Sinh viên sẽ được cộng điểm nếu sử dụng thuật toán Dijkstra với cấu trúc dữ liệu Heap và xử lý với số nguyên lớn.

|  |  |
| --- | --- |
| BYTELAND.IN | BYTELAND.OUT |
| 4 4  1 2 2  1 3 10  1 4 5  2 3 6  1 3 | 8 |