**ĐỀ NGÀY 6 – 12 - 2021**

**Bài 1. Bảo vệ**

Cho 1 cây có N đỉnh. Bạn cần bảo vệ cây này bằng cách đặt các khẩu súng vào các đỉnh sao cho thỏa mãn điều kiện: Mỗi đỉnh hoặc là có 1 khẩu súng đặt vào nó, hoặc là có một đỉnh kề của nó (có đường nối trực tiếp) có 1 khẩu súng đặt vào. Tìm cách đặt sao cho dùng ít khẩu súng nhất.

**Input:** Guard.inp

Dòng đầu tiên là số N (1 <= N <= 1000).

N – 1 dòng tiếp theo, mỗi dòng gồm 2 số u và v cho biết có đường nối trực tiếp giữa 2 đỉnh u và v.

**Output:** Guard.out

Ghi ra 1 số duy nhất là số lượng khẩu súng ít nhất cần dùng.

|  |  |
| --- | --- |
| Guard.inp | Guard.Out |
| 2  1 2 | 1 |
| 4  4 2  2 1  1 3 | 2 |

**Bài 2. City**

Một thành phố lớn nổi tiếng với các danh lam thắng cảnh thu hút rất nhiều khách du lịch. Người ta đến thành phố với mục đích chính là tham quan các thắng cảnh và những địa điểm nổi tiếng của thành phố. Khách du lịch là những người rất bận rộn vì vậy không ai muốn thăm quá một lần một loại danh lam thắng cảnh. Ví dụ, để về kể chuyên là tôi đã thăm viện bảo tàng thì chỉ cần vào một viện bảo tàng nào đó là đủ.

Tất cả ***n*** địa điểm nổi tiếng của thành phố đều nằm dọc trên đường phố chính, địa điểm thứ ***i*** thuộc loại ***ti*** (***i*** = 1 ÷ ***n***). Các tour du lịch bắt đầu từ một danh lam nào đó và kết thúc ở một danh lam khác. Phụ thuộc vào nơi dừng chân và sau đó là nơi thuận tiện trở về mỗi khách du lịch đăng ký một tour tham quan riêng. Khách thứ ***j*** đăng ký tour [***lj***, ***rj***] (bắt đầu từ điểm ***lj*** đến hết điểm ***rj***). Hướng dẫn viên chỉ cần dừng lại giới thiệu kỹ mỗi loại danh lam một lần trên tuyến du lịch. Dĩ nhiên khách muốn được giới thiệu càng nhiều địa điểm càng tốt, nhưng thể loại phải khác nhau từng đôi một.

Hãy xác định số địa điểm mỗi khách du lịch được giới thiệu.

***Dữ liệu:*** Vào từ file văn bản CITY.INP:

* Dòng đầu tiên chứa số nguyên ***n*** (1 ≤ ***n*** ≤ 105),
* Dòng thứ 2 chứa ***n*** số nguyên ***t1***, ***t2***, . . ., ***tn*** (1 ≤ ***ti*** ≤ 109, ***i*** = 1 ÷ ***n***),
* Dòng thứ 3 chứa số nguyên ***m*** – số khách du lịch (1 ≤ ***m*** ≤ 105),
* Dòng thứ ***j*** trong ***m*** dòng sau chứa 2 số nguyên ***lj*** và ***rj*** (1 ≤ ***lj*** ≤ ***rj*** ≤ ***n***, ***j*** = 1 ÷ ***m***).

***Kết quả:*** Đưa ra file văn bản CITY.OUT:

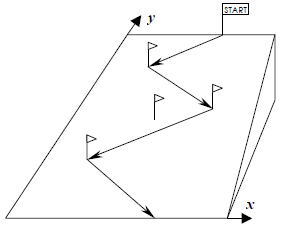
* ***m*** số nguyên, mỗi số trên một dòng. Số nguyên trên dòng thứ ***j*** xác định số địa điểm khách thứ ***j*** được giới thiệu.

***Ví dụ:***

|  |  |
| --- | --- |
| **CITY.INP** | **CITY.OUT** |
| 5  1 1 2 2 1  5  1 5  1 2  2 3  3 4  1 1 | 2  1  2  1  1 |

**Ràng buộc:** 50% test có n, m <= 1000

**Bài 3. SKIING**

Có 1 vận động viên trượt tuyết từ trên đồi xuống. Trên cái dốc có N cái cọc và vận động viên đang tính toán chiến thuật trượt xuống dựa trên mấy cái cọc ấy. Anh ta có thể bắt đầu từ bất kỳ vị trí nào ở vạch xuất phát (vạch này tọa độ y = 100000) và kết thúc ở bất kỳ vị trí nào ở vạch đích (tọa độ y = 0). Đường đi phải là 1 đường gấp khúc có đỉnh là các cọc. Tọa độ y phải ở trạng thái giảm dần ở mọi thời điểm. Anh ta muốn đường đi của mình chạm nhiều cọc nhất có thể, và mỗi khi chạm cọc thì đường đi của anh ta lại đổi hướng. Anh ta có thể đi qua 1 cọc mà không chạm vào nó (cũng không đổi hướng). Xem hình vẽ miêu tả một trường hợp.

Tính toán một đường đi tối ưu cho anh ta.

**Input:** Skiing.inp

Dòng đầu tiên chứa số n là số cọc (không quá 50000). N dòng tiếp theo mỗi dòng miêu tả tọa độ một cọc. Không có 2 cọc nào có cùng tọa độ x hoặc tọa độ y và các tọa độ x, y nằm trong khoảng từ 1 tới 99999.

**Output:** Skiing.out

Ghi ra số cọc nhiều nhất có thể chạm trên đường đi của vận động viên.

|  |  |
| --- | --- |
| Skiing.Inp | Skiing.Out |
| 5  5 2  6 5  1 1  4 3  2 4 | 4 |
| 4  1 6  3 4  5 2  4 1 | 3 |

**Bài 4. Hợp đồng**

Bác thợ mộc Sam nhận được N hợp đồng. Xem xét các hợp đồng, bác thấy mình thiếu mất M ô tô để thực hiện chúng. Tuy vậy, không phải mọi hợp đồng đều cần thêm M ô tô, nhưng việc thực hiện bất kỳ hợp đồng nào cũng cần thêm ít nhất một ô tô.

Để thực hiện hợp đồng, bác Sam có thể mua hoặc thuê thêm cho đủ xe. Giá thuê xe để thực hiện một hợp đồng phụ thuộc vào thời điểm thuê, tức là phụ thuộc vào trình tự thực hiện hợp đồng. Giá mua xe không phụ thuộc vào thời điểm mua và xe đã mua có thể sẽ được sử dụng vào việc thực hiện các hợp đồng tiếp theo mà không cần phải chi thêm khoản phụ phí nào.

Nếu giá thuê hoặc mua xe quá cao, bác Sam có thể từ chố ký hợp đồng mà không phải mất một khoản phí nào (dĩ nhiên – cũng không thu được thêm gì cả).

Háy giúp bác Sam quyết định từ chối hợp đồng nào, mua thêm bao nhiêu xe và khi nào nên thuê xe để có tổng lợi nhuận cao nhất.

Ví dụ, có 2 hợp đồng ***O1*** và ***O2*** và thiếu tổng cộng 3 xe (N = 2, M = 3). Giá mua các xe M1, M2, M3 tương ứng là 50, 80 và 110. Để thực hiện hợp đồng *O1* cần các xe M1 và M2, nếu thuê thì mất tương ứng là 30 và 20. Để thực hiện hợp đồng O2 cần các xe M1 và M3 với chi phí tương ứng là 40 và 80.

Có hai phương án để đạt được lợi nhuận tối đa 50:

* Bỏ hợp đồng O2, chỉ thực hiện hợp đồng O1, thuê cả 2 xe M1 và M2,
* Thực hiện cả hai hợp đồng, mua xe M1 và thuê các xe M2, M3.

***Dữ liệu***: Vào từ file văn bản CONTRACT.INP:

* Dòng đầu tiên chứa 2 số nguyên N và M (1 ≤ N, M ≤ 1 200),
* Nhóm thứ i trong N nhóm dữ liệu sau mô tả hợp đồng Oi:
  + Dòng đầu tiên trong nhóm chứa 2 số nguyên vi – giá trị hợp đồng và mi – số xe cần thiết (1 ≤ vi ≤ 5 000, 1 ≤ mi ≤ M),
  + Mỗi dòng trong mi dòng tiếp theo chứa 2 số nguyên j – số của xe ô tô và rij - chi phí thuê nó cho hợp đồng Oi (1 ≤ rij ≤ 20 000),
* Dòng thứ i trong M dòng cuối cùng chứa số nguyên si – giá mua xe i (1 ≤ si ≤ 20 000).

***Kết*** ***quả***: Đưa ra file văn bản CONTRACT.OUT một số nguyên duy nhất – lợi nhuận tối đa.

***Ví dụ***:

|  |  |  |
| --- | --- | --- |
| **CONTRACT.INP** |  | **CONTRACT.OUT** |
| 2 3  100 2  1 30  2 20  100 2  1 40  3 80  50  80  110 |  | 50 |