

### TỔNG QUAN ĐỀ THI

| Bài | Tên bài            | Tên file chương trình | Hạn chế thời gian | Hạn chế bộ nhớ |
|-----|--------------------|-----------------------|-------------------|----------------|
| 1   | Nối điểm đen trắng | BWPOINTS.*            | 1 giây            | 256 M          |
| 2   | Chu trình          | CIRCUIT.*             | 1 giây            | 256 M          |
| 3   | Lịch học           | SCHEDULE.*            | 1 giây            | 256 M          |
| 4   | Tham ăn            | SMARTDOG.*            | 1 giây            | 256 M          |
| 5   | Tuyển dụng         | EMPLOY.*              | 1 giây            | 256 M          |

Tên Dấu \* được thay thế bởi PAS hoặc CPP của ngôn ngữ lập trình được sử dụng tương ứng là Pascal hoặc C++.

### Lập trình giải các bài toán sau đây:

#### NỐI ĐIỂM ĐEN TRẮNG

Trên trục số thực cho  $n$  điểm đen và  $n$  điểm trắng hoàn toàn phân biệt. Các điểm đen có tọa độ nguyên  $a_1, a_2, \dots, a_n$  còn các điểm trắng có tọa độ nguyên  $b_1, b_2, \dots, b_n$ . Người ta muốn chọn ra  $k$  điểm đen và  $k$  điểm trắng để nối mỗi một điểm đen với một điểm trắng sao cho  $k$  đoạn thẳng tạo được đôi một không có điểm chung.

**Yêu cầu:** Cho tọa độ của  $n$  điểm đen và tọa độ của  $n$  điểm trắng, hãy tìm giá trị  $k$  lớn nhất thỏa mãn yêu cầu nêu trên.

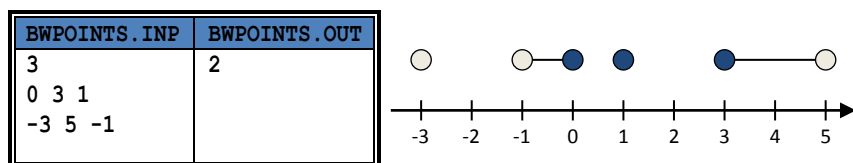
**Dữ liệu:** Vào từ file văn bản BWPOINTS.INP:

- Dòng thứ nhất chứa số nguyên dương  $n \leq 10^5$
- Dòng thứ hai chứa các số  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 10^9, i = 1, 2, \dots, n$ )
- Dòng thứ ba chứa các số  $b_1, b_2, \dots, b_n$  ( $|b_i| \leq 10^9, i = 1, 2, \dots, n$ )

Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản BWPOINTS.OUT một số nguyên duy nhất là số  $k$  lớn nhất tìm được.

**Ví dụ**



Ràng buộc: 50% số tests ứng với 50% số điểm của bài có  $1 \leq n \leq 100$ .

## CHU TRÌNH

Cho đồ thị có hướng  $G = (V, E)$  có  $n$  đỉnh đánh số từ 1 tới  $n$  và  $m$  cung đánh số từ 1 tới  $m$ . Cung thứ  $i$  nối từ đỉnh  $u_i$  tới đỉnh  $v_i$ . Giữa hai đỉnh có thể có nhiều cung nối chúng và có thể có cung nối từ 1 đỉnh tới chính nó.

Một dãy có thứ tự các cung  $e_1, e_2, \dots, e_k$  được gọi là một *chu trình trên  $G$*  nếu  $k \geq 1$  và tồn tại dãy đỉnh  $v_1, v_2, \dots, v_k, v_{k+1} = v_1$  sao cho cung  $e_i$  nối từ  $v_i$  tới  $v_{i+1}$  ( $\forall i = 1, 2, \dots, k$ ).

**Yêu cầu:** Tìm một chu trình trên  $G$ .

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

- Dòng 1 chứa hai số nguyên dương  $n, m \leq 10^5$
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa hai số nguyên dương  $u_i, v_i$

**Kết quả:** Ghi ra file văn bản CIRCUIT.OUT

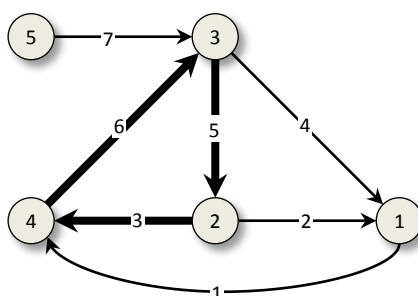
- Dòng 1: Ghi từ YES nếu đồ thị  $G$  có chu trình, ghi từ NO nếu đồ thị  $G$  không có chu trình. Các ký tự đều là phải là CHỮ HOA.
- Nếu đồ thị  $G$  có chu trình, dòng 2 ghi số hiệu các cung theo đúng thứ tự trên chu trình tìm được

*Các số trên một dòng của input/output files được/phải ghi cách nhau ít nhất một dấu cách*

*Nếu có nhiều chu trình trên đồ thị, chỉ đưa ra một chu trình*

**Ví dụ**

| CIRCUIT | CIRCUIT |
|---------|---------|
| 5 7     | YES     |
| 1 4     | 3 6 5   |
| 2 1     |         |
| 2 4     |         |
| 3 1     |         |
| 3 2     |         |
| 4 3     |         |
| 5 3     |         |



## LỊCH HỌC

Chương trình học của một trường đại học có  $n$  môn đánh số từ 1 tới  $n$ , môn học thứ  $i$  cần học trong  $t_i$  học kỳ. Có  $m$  quan hệ giữa các môn học, quan hệ thứ  $j$  cho biết sinh viên bắt buộc phải học xong môn  $u_j$  trước khi bắt đầu học môn  $v_j$ . Chương trình đào tạo được cho hợp lý để sinh viên có thể hoàn thành hết tất cả các môn học, các học kỳ được đánh số từ 1 theo trình tự thời gian. Sinh viên có thể học nhiều môn trong cùng một học kỳ miễn là không vi phạm những mối quan hệ giữa các môn học.

**Yêu cầu:** Hãy lập một lịch học để sinh viên có thể hoàn thành hết tất cả các môn một cách nhanh nhất.

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

- Dòng 1 chứa hai số nguyên dương  $n \leq 10^5$ ;  $m \leq 2 \cdot 10^5$
- Dòng 2 chứa  $n$  số nguyên dương  $t_1, t_2, \dots, t_n$  ( $t_i \leq 10^5, \forall i$ )
- $m$  dòng tiếp theo, dòng thứ  $j$  chứa hai số nguyên dương  $u_j, v_j$

**Kết quả:** Ghi ra file văn bản SCHEDULE.OUT

- Dòng 1: Ghi số học kỳ ít nhất để hoàn thành tất cả các môn
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi số hiệu học kỳ bắt đầu học môn  $i$

**Các số trên một dòng của Input/Output Files được/phải ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

| SCHEDULE . INP | SCHEDULE . OUT |
|----------------|----------------|
| 5 6            | 12             |
| 5 4 3 2 1      | 8              |
| 5 2            | 4              |
| 5 1            | 1              |
| 3 2            | 8              |
| 3 4            | 1              |
| 2 1            |                |
| 2 4            |                |

## THAM ĂN

Tiếp theo chiến lược “quy hoạch động”, Bờm huấn luyện cho chú chó của mình chiến lược “tham ăn” trong một sân chơi được biểu diễn bởi mặt phẳng trục chuẩn  $Oxy$ . Ban đầu chú chó xuất phát ở điểm  $(0,0)$  và nó phải đứng im cho tới khi được gọi. Trò chơi diễn ra trong  $n$  lượt, lượt thứ  $i$  của trò chơi diễn ra như sau:

Bờm di chuyển đến vị trí  $(x_i, y_i)$ , cầm  $c_i$  cái bánh và gọi chú chó. Chú chó có quyền đứng im hoặc di chuyển theo các phương song song theo trục tọa độ để đến chỗ Bờm nếu độ dài quãng đường di chuyển không vượt quá  $\Delta$ . Nếu chú chó có thể đi được đến chỗ Bờm và quyết định di chuyển, nó sẽ được thưởng toàn bộ  $c_i$  cái bánh, ngược lại nó sẽ phải đứng nhìn Bờm ăn hết luôn  $c_i$  cái bánh đó. Hết lượt chơi này chú chó lại phải đứng im và trò chơi tiếp tục ở lượt  $i + 1$ .

**Yêu cầu:** Cho biết trước các tọa độ  $(x_i, y_i)$  và số bánh  $c_i$  tại các lượt chơi, hãy giúp chú chó tối nghiệp của Bờm kiếm được nhiều bánh nhất

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

- Dòng 1 chứa hai số nguyên dương  $n, \Delta$
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa ba số nguyên dương  $x_i, y_i, c_i$

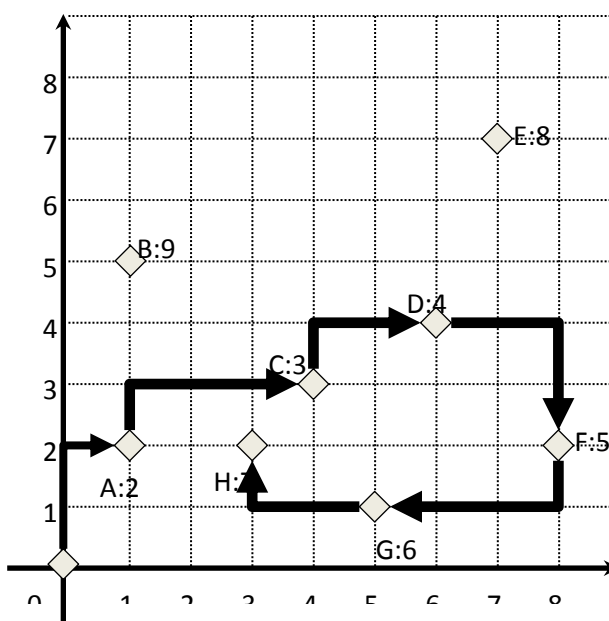
Ràng buộc:  $n \leq 10^5$ , tất cả các số còn lại trong file dữ liệu đều là số nguyên dương  $\leq 10^3$

**Kết quả:** Ghi ra file văn bản SMARTDOG.OUT một số nguyên duy nhất là số bánh chú chó kiếm được trong trò chơi theo phương án của bạn

### Ví dụ

Đây là ví dụ với 8 lượt chơi và vị trí của Bờm tại 8 lượt chơi đó lần lượt là A, B, C, D, E, F, G, H. Trong phương án tối ưu chú chó chỉ phải bỏ 9 bánh tại điểm B và 8 bánh tại điểm E.

| SMARTDOG . INP  | SMARTDOG . OUT |
|---|----------------|
| 8 4<br>1 2 2<br>1 5 9<br>4 3 3<br>6 4 4<br>7 7 8<br>8 2 5<br>5 1 6<br>3 2 7 | 27             |



## TUYỂN DỤNG

Một siêu thị cần tuyển một số nhân viên bán hàng. Giờ làm việc trong mỗi ngày được tính từ thời điểm 0 tới thời điểm  $t$  ( $[0, t]$ ). Có  $n$  ứng viên đánh số từ 1 tới  $n$ . Ứng viên thứ  $i$  chỉ có thể làm từ thời điểm  $a_i$  tới thời điểm  $b_i$  trong ngày ( $[a_i, b_i]$ ) nếu được tuyển dụng và ứng viên đó yêu cầu mức lương mỗi ngày là  $c_i$ .

**Yêu cầu:** Hãy giúp siêu thị tuyển một số nhân viên bán hàng trong số các ứng viên sao cho: Bất kỳ thời điểm nào trong giờ làm việc cũng có ít nhất một nhân viên bán hàng và tổng tiền lương phải trả trong mỗi ngày cho các nhân viên là ít nhất.

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

- Dòng 1 chứa số nguyên dương  $n \leq 10^5$  và số nguyên dương  $t \leq 10^9$
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa ba số nguyên  $a_i, b_i, c_i$  ( $0 \leq a_i < b_i \leq t; 1 \leq c_i \leq 10^9$ )

**Kết quả:** Vào từ file văn bản EMPLOY.OUT

- Dòng 1 ghi tổng tiền lương phải trả mỗi ngày cho các nhân viên theo phương án tìm được.
- Dòng 2 ghi chỉ số những ứng viên được chọn trong phương án tìm được theo thứ tự tùy ý.

*Các số trên một dòng của input/output files được phải ghi cách nhau ít nhất một dấu cách*

*Dữ liệu vào luôn đảm bảo tồn tại phương án tuyển dụng theo yêu cầu đặt ra*

*Nếu có nhiều phương án cùng tối ưu, chỉ đưa ra một phương án*

Ví dụ

| EMPLOY . INP | EMPLOY . OUT |
|--------------|--------------|
| 5 8          | 100          |
| 0 2 10       | 2 4 5        |
| 0 4 30       |              |
| 1 7 90       |              |
| 3 7 60       |              |
| 7 8 10       |              |

