

## TỔNG QUAN ĐỀ THI

Bài	Tên bài	File chương trình	File dữ liệu	File kết quả	Điểm
1	Ước số	SOU.*	SOU.INP	SOU.OUT	7
2	Thư viện sách	SACH.*	SACH.INP	SACH.OUT	7
3	Con đường bội của 2	BOI2.*	BOI2.INP	BOI2.OUT	6

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

### Bài 1: Ước số

Nam là một người bạn của Định đang tìm cách giải một bài toán liên quan tới số tự nhiên. Lúc này, Nam đang rất cần sự giúp đỡ của Định. Thử thách lần này là một số tự nhiên  $N(N \leq 10^6)$ , với  $N!$  ( $N$  giai thừa) hãy tìm ra số các ước số và tổng các ước số của nó. Vì  $N!$  đã cho là rất lớn nên việc tìm thủ công là không thể. Chính vì vậy, Định cần một thuật toán để cài đặt vào máy tính và nhờ máy tính tìm giúp.

**Yêu cầu:** Bạn hãy giúp Định tìm ra: Số các ước số và tổng các ước số của  $N!$ .

**Dữ liệu** vào từ file SOU.INP gồm

- Một số tự nhiên  $N$ .

**Kết quả** ghi vào file SOU.OUT có cấu trúc như sau:

- Dòng thứ nhất là số các ước số của  $N!$ ;
- Dòng thứ hai là tổng các ước số của  $N!$ ;

*Lưu ý vì số các ước và tổng các ước số của  $N!$  có thể rất lớn nên kết quả là số dư của  $10^7+7$ .*

**Ví dụ:**

SOU.INP	SOU.OUT
4	8
	60

**Ràng buộc:**

- *Sub1:* Có 20% số test ứng với 20% số điểm với  $N \leq 100$ ;

- Sub2: Có 40% số test ứng với 40% số điểm với  $N \leq 10^4$ ;
- Sub3: Có 40% số test còn lại ứng với 40% số điểm với  $N \leq 10^6$ .

## Bài 2: Thư viện sách

Trong hệ thống thư viện tỉnh XYZ, cần thiết kế một kệ sách để tiết kiệm không gian. Thư viện có  $n$  quyển sách, quyển thứ  $i$  có chiều cao là  $h_i$ , chiều rộng  $t_i$ . Thư viện muốn thiết kế một giá sách gồm 3 tầng để có thể chứa hết tất cả  $n$  quyển sách.

Giả sử  $n$  quyển sách được phân thành 3 tập không rỗng  $S_1, S_2, S_3$  (các quyển sách thuộc tập  $S_i$  được xếp vào tầng  $i$ ) thì cần giá sách chiếm diện tích bằng:

$$\left( \sum_{j=1}^3 \max_{i \in S_j} (h_i) \right) \times \left( \max_{j=1}^3 \sum_{i \in S_j} t_i \right)$$

**Yêu cầu:** Bạn hãy giúp thư viện tìm cách phân  $n$  quyển thành 3 tập khác rỗng để kệ sách chiếm diện tích nhỏ nhất.

**Dữ liệu** vào từ file SACH.INP gồm

- Dòng đầu tiên là số  $n$ ;
- $N$  dòng tiếp theo mỗi dòng 2 số  $h_i$  và  $t_i$  ( $150 \leq h_i \leq 300, 5 \leq t_i \leq 30$ ).

**Kết quả** ghi vào file SACH.OUT ghi số diện tích nhỏ nhất của giá sách.

**Ví dụ:**

SACH.INP	SACH.OUT
4	18000
220 29	
195 20	
200 9	
180 30	

## Ràng buộc:

- Sub1: Có 30% số test ứng với 30% số điểm với  $n \leq 10$ ;
- Sub2: Có 30% số test ứng với 30% số điểm với  $n \leq 20$ ;
- Sub3: Có 40% số test còn lại ứng với 40% số điểm với  $n \leq 70$ .

## Bài 3: Con đường bội của 2

Tom sống ở Disneyland, một quốc gia có rất nhiều thành phố và đường đi hai chiều nối các thành phố đó. Mỗi đường hai chiều có chiều dài **đôi một khác nhau**, và mỗi chiều dài là *bội của 2* (nghĩa là tăng lên một số mũ). Tom có thể đến bất cứ thành phố nào từ bất kỳ thành phố nào khác.

Với bản đồ của Disneyland, bạn hãy giúp Tom xác định tổng khoảng cách tối thiểu giữa mỗi cặp thành phố? In câu trả lời của bạn dưới dạng biểu diễn nhị phân.

**Dữ liệu** vào từ file BOI2.INP gồm

- Dòng đầu chứa 2 số nguyên  $N$ (số thành phố) và  $M$ (số đường đi 2 chiều nối giữa chúng).
- Dòng thứ  $i$  của  $M$  dòng tiếp theo chứa các giá trị tương ứng  $A_i, B_i, C_i$  ( $C_i \leq M$ ) có nghĩa là có đường đi hai chiều giữa thành phố 'A<sub>i</sub>' và thành phố 'B<sub>i</sub>' với độ dài là  $2^{C_i}$ .

**Kết quả** ghi vào file BOI2.OUT

- Ghi ra tổng khoảng cách tối thiểu giữa mỗi cặp thành phố dưới dạng biểu diễn nhị phân .

**Ví dụ:**

BOI2.INP	BOI2.OUT
5 6	1000100
1 3 5	
4 5 0	
2 1 3	
3 2 1	
4 3 4	
4 2 2	

**Ràng buộc:**

- *Sub1*: Có 30% số test ứng với 30% số điểm với  $N \leq 30, M \leq 40$  và  $1 \leq A_i, B_i \leq N, A_i \neq B_i$  ;
- *Sub2*: Có 30% số test ứng với 30% số điểm với  $N \leq 1000; M \leq 1000$  và  $1 \leq A_i, B_i \leq N, A_i \neq B_i$  ;

- *Sub3: Có 40% số test còn lại ứng với 40% số điểm với  $N \leq 10^5$ ;  $M \leq 2 \cdot 10^5$  và  $1 \leq A_i$ ,  $B_i \leq N$ ,  $A_i \neq B_i$ .*

----- **HẾT** -----