

## Bài A. TWOLAYER

File dữ liệu vào: **stdin**  
File kết quả: **stdout**  
Hạn chế thời gian: 1 giây

Huy đang muốn mua tặng Giang một món quà nhân ngày sinh nhật. Nơi Huy sống có  $n$  thành phố đánh số 1 đến  $n$ . Món quà mà Huy muốn mua, ở thành phố  $i$  được bán với giá  $c_i$  đồng. Hệ thống giao thông công cộng ở đây chủ yếu là tàu điện ngầm. Mỗi tuyến có hai định mức giá vé, là giá vé đi lần đầu của tuyến này và giá vé đi không phải lần đầu của tuyến này. Có  $m$  tuyến tàu điện như vậy, tuyến thứ  $i$  di chuyển hai chiều giữa thành phố  $u_i$  và  $v_i$ , hai định mức giá vé là  $w_i$  và  $l_i$  đồng ( $l_i \leq w_i$ ). Điều này có nghĩa, nếu bạn mua vé tuyến thứ  $i$  (bất kỳ chiều nào) thì bạn phải trả  $w_i$  đồng, và trong ngày đó nếu muốn mua tiếp các vé của tuyến thứ  $i$  (bất kỳ chiều nào), bạn sẽ chỉ mất  $l_i$  đồng cho mỗi vé

Nhà Huy ở thành phố 1, nhà Giang ở thành phố  $n$ . Huy muốn tìm một lộ trình trong ngày, xuất phát từ nhà mình, mua quà ở một thành phố bất kỳ nào đó và kết thúc ở nhà Giang - tất nhiên rồi. Hãy giúp anh ấy tìm lộ trình sao cho tổng số tiền phải dùng là ít nhất có thể.

### Dữ liệu vào

- Dòng đầu chứa  $n$   $m$
- Dòng tiếp theo chứa  $n$  số tự nhiên:  $c_1, c_2, \dots, c_n$
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa:  $u_i$   $v_i$   $w_i$   $l_i$

### Kết quả

Ghi tổng số tiền nhỏ nhất có thể

### Ví dụ

stdin	stdout
5 6 10 10 1 10 10 1 2 2 2 2 3 1 1 3 4 1 1 4 5 2 2 1 5 1 1 2 5 2 0	6

### Giải thích

Huy sẽ đi 1 -> 5 -> 2 -> (3) -> 2 -> 5

### Hạn chế

- $n, m \leq 10^5$ .  $0 \leq w_i, l_i, c_i \leq 10^9$
- 20% test với  $l_i = w_i$
- 20% test tiếp theo có  $l_i = 0$
- 30% test tiếp theo có  $n \leq 1000$
- 30% test tiếp theo không có ràng buộc bổ sung

## Bài B. MXOR

File dữ liệu vào: **stdin**  
File kết quả: **stdout**  
Hạn chế thời gian: 1 giây

Long có một dãy số nguyên không âm  $a = a_1, a_2, \dots, a_n$  và muốn tìm ra  $(i, j)$ ,  $1 \leq i < j \leq n$ , sao cho  $a_i \wedge a_j$  lớn nhất có thể. Ở đây  $\wedge$  là phép toán xor, hay còn gọi là phép hoặc triết tiêu. Dữ liệu đảm bảo cặp  $(i, j)$  tối ưu là duy nhất.

Long nổi tiếng với tính tình cổ quái, cậu muốn được giúp nhưng lại không muốn tiết lộ dãy  $a$ . Thay vào đó, bạn có thể trao đổi với Long bằng các hàm sau:

- `int get_n()`: Trả ra số phần tử của dãy  $a$
- `int max_xor(vector<int> &I, vector<int> &J)` (hoặc `int max_xor(int[] I, int[] J)` với Java): Trả ra  $\max_{i \in I, j \in J} a_i \wedge a_j$  với  $I$  và  $J$  là hai tập con không giao nhau của tập  $\{1, 2, \dots, n\}$
- `void answer(int i, int j)`: Trả lời Long

Long sẽ trả lời không quá 33 câu hỏi dạng `max_xor(I, J)`. Nếu bạn hỏi quá nhiều, hỏi với  $I$  và  $J$  không phải tập con của  $\{1, 2, \dots, n\}$  hoặc hỏi với  $I$  và  $J$  có phần tử chung thì bài của bạn sẽ bị chấm sai.

Để sử dụng các hàm trên, với C++ bạn cần khai báo `#include "MXORLIB.h"` ở đầu chương trình, sau đó các hàm đó có thể được gọi ở bất kỳ đâu trong chương trình của bạn. Xem file `MXORLIB.h` và `MXOR.cpp` trong mục đính kèm để hiểu rõ hơn cách tương tác. Lưu ý đây là thư viện ví dụ, có thể khác với thư viện dùng để chấm bài. Trình chấm đảm bảo rằng nếu chương trình của bạn biên dịch được với thư viện ví dụ thì cũng biên dịch được trên hệ thống (để `MXORLIB.h` và chương trình của bạn vào cùng một thư mục rồi biên dịch như bình thường).

Đối với Java, hệ thống cung cấp sẵn một class tên là `MXORLIB` và bạn không cần phải khai báo gì thêm. Các hàm trên được để static, thí sinh gọi `MXORLIB.get_n()`, `MXORLIB.max_xor(I, J)`, `MXORLIB.answer(i, j)` để tương tác.

## Hạn chế

- $n \leq 10^5$ ,  $0 \leq a_i \leq 10^9$  trong tất cả các test
- Có 20% số test với  $n \leq 17$
- Có 30% số test với  $17 < n \leq 10^4$
- Có 50% số test với  $10^4 < n \leq 10^5$

## Bài C. SPGM

File dữ liệu vào: **stdin**  
File kết quả: **stdout**  
Hạn chế thời gian: 1 giây

Hai người S và P cùng nhau chơi một trò chơi và nhờ GM làm trọng tài. Trò chơi bắt đầu bởi một số nguyên dương  $N$  là số được thống nhất bởi ba người họ. Trọng tài sẽ chọn ra hai số nguyên dương KHÁC nhau trong khoảng từ 1 đến  $N$ , sau đó nói cho S biết tổng và P biết tích của hai số đó. Hai người S và P sẽ luân phiên nhau cố gắng đoán ra hai số này, S đoán trước. Ví dụ với  $N = 10$ , GM chọn 3 và 6 thì trò chơi sẽ diễn ra như sau:

- S: "Tao không biết"
- P: "Tao không biết"
- S: "Tao không biết"
- P: "Tao không biết"
- S: "Tao biết rồi, đó là 3 và 6"

Hoặc với  $N = 15$ , GM chọn 3 và 5 thì trò chơi sẽ diễn ra như sau:

- S: "Tao không biết"
- P: "Tao không biết"
- S: "Tao không biết"
- P: "Thôi dẹp đi, có đoán đến sáng mai cũng không ra đâu"

Cho biết  $N$  là số mà cả ba người đều biết,  $x$  và  $y$  là hai số mà GM đã chọn (tức là GM sẽ nói cho S biết giá trị của  $x + y$  và nói cho P biết giá trị của  $x \times y$ ), hãy đếm số lần mà câu "Tao không biết" được nói ra. Biết rằng cả hai người chơi đều rất thông minh và trung thực

### Dữ liệu vào

- Dòng đầu chứa số lượng test:  $T$
- $T$  dòng tiếp theo mỗi dòng chứa:  $N \ x \ y$

### Kết quả

- Gồm  $T$  dòng ghi số lần mà câu "Tao không biết" được nói ra tương ứng cho  $T$  test

### Ví dụ

stdin	stdout
2	4
10 3 6	3
15 3 5	

### Hạn chế

- $2 \leq N \leq 200$ ,  $0 \leq M \leq 100$ ,  $1 \leq T \leq 10$
- Có ít nhất 60% số test mà S hoặc P sẽ đoán ra được  $x, y$

## Bài D. BSCOL

File dữ liệu vào: **stdin**  
File kết quả: **stdout**  
Hạn chế thời gian: 1 giây

Ngày xưa có một cô gái rất yêu thiên nhiên. Cô đã thu thập được rất nhiều hạt cườm thiên nhiên với đầy đủ màu sắc, cô sẽ sử dụng chúng để làm một cái vòng cổ gồm không quá  $n$  hạt (tất nhiên là ít nhất 1 hạt). Vì yêu màu xanh lá cây nên cô quyết định sẽ dùng các màu ... xanh đỏ tím vàng (ai mà hiểu nổi :v). Đặc biệt hơn, cô không muốn màu xanh và màu đỏ kề nhau, cũng không muốn màu tím và màu vàng kề nhau, đồng thời, hai hạt kề nhau phải khác màu

Bạn trai cô không quan tâm lắm đến thẩm mỹ, anh ta tò mò về số cách khác nhau mà cô ấy có thể tạo thành cái vòng cổ đó. Hãy giúp anh ta tính toán để nhanh chóng lấy được lòng cô ấy! Lưu ý là hai vòng được coi là giống nhau nếu sau một phép xoay chúng trùng hằn với nhau

### Dữ liệu vào

- Gồm một số tự nhiên duy nhất:  $n$

### Kết quả

- In ra phần dư của số cách chia khi chia cho  $10^9 + 7$

### Ví dụ

stdin	stdout
2	8
6	42

### Hạn chế

- Subtask 0:  $n \leq 10$
- Subtask 1:  $n \leq 1000$
- Subtask 2:  $n \leq 10^5$