

## TÍCH LỚN NHẤT

Cho dãy  $A$  gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  và một số nguyên dương  $k \leq n$ .

**Yêu cầu:** Hãy chọn ra trong dãy này đúng  $k$  phần tử sao cho tích của  $k$  phần tử này lớn nhất.

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

- ✿ Dòng 1 chứa số  $T$  là số test
- ✿  $T$  nhóm dòng tiếp theo, mỗi nhóm gồm 2 dòng ứng với một test:
  - ✿ Dòng thứ nhất chứa hai số nguyên dương  $n, k$  ( $1 \leq k \leq n \leq 10^5$ );
  - ✿ Dòng thứ hai chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $\forall i: |a_i| \leq 10^9$ )

*Các số trên một dòng của input file được ghi cách nhau bởi dấu cách, số lượng các số trong một file dữ liệu không quá  $2 \cdot 10^5$*

**Kết quả:** Ghi ra file văn bản PRODUCT.OUT một số nguyên duy nhất là số dư của tích  $k$  phần tử được chọn khi chia cho 123456789

**Ví dụ:**

PRODUCT.INP	PRODUCT.OUT
3	60
5 3	1
1 2 3 4 5	3
6 4	
-1 -1 -1 -1 0 9	
5 3	
-1 -1 -1 2 3	

## KHOẢNG CÁCH LỚN NHẤT

Cho dãy số nguyên  $A = (a_1, a_2, \dots, a_n)$ . Với số nguyên  $x$ , định nghĩa khoảng cách từ  $x$  tới dãy  $A$  là:

$$\min_{i=1,2,\dots,n} \{|x - a_i|\}$$

**Yêu cầu:** Tìm số nguyên  $x \in [L, R]$  sao cho khoảng cách từ  $x$  tới dãy  $A$  là lớn nhất. Nếu có nhiều giá trị  $x$  có cùng khoảng cách tới  $A$  và đều là lớn nhất, cần chỉ ra giá trị  $x$  lớn nhất.

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

- Dòng 1 chứa ba số nguyên  $n, L, R$  ( $1 \leq n \leq 10^5; -2^{63} \leq L \leq R < 2^{63}$ )
- Dòng 2 chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $\forall i: -2^{63} \leq a_i < 2^{63}$ )

**Kết quả:** Ghi ra file văn bản MAXDIS.OUT một số nguyên duy nhất là giá trị số  $x$  tìm được

MAXDIS.INP	MAXDIS.OUT
4 3 8	7
2 4 6 8	

## TỔNG XOR

Cho dãy số nguyên dương  $A = (a_1, a_2, \dots, a_n)$ , ta gọi tổng XOR của nó là

$$a_1 \text{ XOR } a_2 \text{ XOR } \dots \text{ XOR } a_n$$

**Yêu cầu:** Chọn một dãy con của  $A$  sao cho tổng XOR của dãy con này là lớn nhất.

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

- ✿ Dòng 1 chứa số nguyên dương  $n \leq 10^5$
- ✿ Dòng 2 chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $\forall i: a_i \leq 10^{18}$ )

**Kết quả:** Ghi ra file văn bản SXOR.OUT **chỉ số** những phần tử được chọn trên một dòng cách nhau bởi dấu cách

**Ví dụ**

SXOR.INP	SXOR.OUT
4	4 2
1 2 4 5	
5	1 3 4 5
14 8 13 6 10	

Giải thích ví dụ 1:  $a[4] \text{ XOR } a[2] = 7$  là tổng XOR lớn nhất có thể chọn được

## HỆ THỐNG GẦN HOÀN HẢO

Một hệ thống  $S$  gồm  $m$  máy biến đổi số được đánh số từ 1 tới  $m$ . Hệ thống thực hiện phép biến đổi trên tập các số nguyên dương từ 1 tới  $n$ . Hoạt động của máy  $i$  được xác định bởi cặp số nguyên dương  $(a_i, b_i)$  ( $1 \leq a_i, b_i \leq n$ ): Máy nhận đầu vào là số nguyên dương  $a_i$  và trả ở đầu ra số nguyên dương  $b_i$ . Như vậy hệ thống  $S$  được mô tả bởi hai dãy số  $A = (a_1, a_2, \dots, a_m)$  và  $B = (b_1, b_2, \dots, b_m)$ .

Ta nói một số nguyên dương  $x$  có thể biến đổi thành số nguyên dương  $y$  nếu  $x = y$  hoặc tồn tại một dãy hữu hạn các số nguyên dương  $x = p_1, p_2, \dots, p_k = y$  sao cho đối với hai phần tử liên tiếp  $p_i, p_{i+1}$  bất kỳ trong dãy, luôn tìm được một trong số các máy đã cho để biến đổi  $p_i$  thành  $p_{i+1}$ .

Hệ thống  $S$  được gọi là gần hoàn hảo nếu với hai số  $a, b$  bất kỳ thuộc tập  $A \cup B$ , hoặc  $a$  có thể biến đổi về  $b$ , hoặc  $b$  có thể biến đổi về  $a$ . Ở đây  $A \cup B$  là ký hiệu tập các phần tử thuộc dãy  $A$  hoặc dãy  $B$ .

**Yêu cầu:** Hãy kiểm tra xem hệ thống  $S$  cho trước có phải là gần hoàn hảo hay không?

**Dữ liệu:** Vào từ file văn bản SPERFECT.INP chứa một số bộ dữ liệu

- Dòng đầu tiên chứa số nguyên dương  $q$  là số bộ dữ liệu
- Tiếp theo là  $q$  nhóm dòng mô tả các bộ dữ liệu:
  - Dòng đầu tiên trong nhóm chứa hai số nguyên dương  $n, m$  ( $1 \leq n, m \leq 10^5$ )
  - $m$  dòng tiếp theo trong nhóm, mỗi dòng chứa một cặp số tương ứng với một máy biến đổi số.

**Kết quả:** Ghi ra  $q$  dòng của file văn bản SPERFECT.OUT: dòng thứ  $i$  (tương ứng với bộ dữ liệu thứ  $i$  trong file dữ liệu vào) chứa thông báo “YES”, nếu hệ thống  $S$  trong bộ dữ liệu tương ứng là gần hoàn hảo, và thông báo “NO” nếu trái lại

**Ví dụ:**

SPERFECT.INP	SPERFECT.OUT
2	YES
6 3	NO
1 3	
2 3	
3 1	
6 2	
1 3	
2 3	

## TRUYỀN TIN TRÊN MẠNG

Cho một mạng gồm  $n$  máy tính đánh số từ 1 tới  $n$  và  $m$  cáp nối đánh số từ 1 tới  $m$ . Cáp nối thứ  $i$  nối giữa hai máy tính  $u_i, v_i$  và cho phép truyền tin giữa hai máy theo cả hai chiều. Hai máy tính  $s$  và  $t$  có thể kết nối được với nhau nếu tồn tại dãy các máy  $s = x_1, x_2, \dots, x_k = t$  sao cho giữa hai máy  $(x_i, x_{i+1})$  có cáp nối chúng ( $\forall i = 1, 2, \dots, k-1$ ). Mạng đảm bảo hai máy bất kỳ có thể kết nối được với nhau. Giữa hai máy tính có thể có nhiều hơn 1 cáp nối.

Ta nói máy  $u$  là xung yếu đối với cặp máy  $(s, t)$  nếu máy như máy  $u$  gặp sự cố (không thể tham gia truyền tin) thì hai máy  $s, t$  không thể kết nối với nhau (tính cả trường hợp  $u = s$  hoặc  $u = t$ ). Tương tự như vậy ta nói một cáp nối là xung yếu đối với cặp máy  $(s, t)$  nếu như cáp này gặp sự cố thì hai máy  $s, t$  không thể kết nối với nhau.

**Yêu cầu:** Cho  $q$  truy vấn, mỗi truy vấn cho bởi một cặp máy khác nhau  $(s, t)$ , hãy cho biết có bao nhiêu máy và cáp nối xung yếu đối với cặp máy  $(s, t)$  đó.

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

- Dòng 1 chứa số nguyên dương  $n \leq 10^5; m \leq 2 \cdot 10^5$
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa hai số nguyên dương  $u_i, v_i$
- Dòng tiếp theo chứa số nguyên dương  $q \leq 10^5$
- $q$  dòng tiếp theo, mỗi dòng chứa chỉ số hai máy khác nhau ứng với một truy vấn

**Kết quả:** Ghi ra file văn bản CNET.OUT  $q$  dòng, mỗi dòng ghi hai số nguyên: Số thứ nhất là số máy xung yếu và số thứ hai là số cáp xung yếu đối với cặp máy trong một truy vấn theo đúng thứ tự trong dữ liệu vào.

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

**Ví dụ**

CNET.INP	CNET.OUT
6 7	2 0
1 2	4 1
1 4	
1 5	
2 4	
3 5	
3 6	
5 6	
2	
2 4	
4 3	

