



VNU Olympiad Informatic

Đề thi ngày 2

Thời gian làm bài: 210 phút

5/13/2020

Tổng quan đề thi

Bài	Mã nguồn	Dữ liệu vào	Dữ liệu ra
PRIMES	primes.cpp pas	primes.inp	primes.out
NMGRAPH	nmgraph.cpp pas	nmgraph.inp	nmgraph.out
TRANS	trans.cpp pas	trans.inp	trans.out
SCI	sci.cpp pas	sci.inp	sci.out

PRIMES

Hoàng và Đông là hai người bạn thân thiết từ lâu rất thích môn toán dù hiện nay hai người làm trong lĩnh vực khác nhưng vẫn thường xuyên đố nhau về những bài toán. Hôm nay Hoàng hứng chí đề xuất Đông một bài toán liên quan đến khái niệm số nguyên tố.

Một số nguyên dương x gọi là số nguyên tố nếu số x chỉ có đúng hai ước khác nhau là 1 và chính nó.

Với một số nguyên dương n , Hoàng viết liên tiếp tất cả các số n nguyên tố đầu tiên theo thứ tự từ nhỏ tới lớn thành một dãy các chữ số. Ví dụ với $n = 10$ Hoàng sẽ nhận được dãy các chữ số: 2357111317192329. Hoàng đưa dãy chữ số đó cho Đông kèm theo một số k và yêu cầu Đông xóa đi đúng k chữ số trong dãy để nhận được số lớn nhất có thể.

Yêu cầu: Với số n và k , hãy giúp Đông tìm ra số lớn nhất có thể nhận được.

INPUT

Dòng đầu chứa số t là số bộ test cần kiểm tra, Dòng thứ i trong t dòng tiếp theo, mỗi dòng chứa hai số n và k .

Đảm bảo rằng k nhỏ hơn độ dài của dãy chữ số.

OUTPUT

In ra t dòng, dòng thứ i ghi kết quả của test thứ i .

Giới hạn: Tổng các số n trong mỗi test không vượt quá 400 000.

primes.inp	primes.out
2	792329
10 10	7719
8 8	

NMGRAPH

Cho một đồ thị vô hướng gồm n đỉnh và m cạnh. Các đỉnh được đánh số từ 1 tới n . Mỗi đỉnh và mỗi cạnh của đồ thị đều có trọng số. Trọng số của đỉnh thứ i là w_i . Cạnh thứ j của đồ thị nối hai đỉnh f_j và t_j có trọng số là c_j . Đồ thị được đảm bảo là liên thông, nói cách khác, luôn tồn tại đường đi giữa hai đỉnh bất kỳ của đồ thị.

Xét một đường đi bất kỳ trên đồ thị, giả sử đường đi đi qua các đỉnh v_1, v_2, \dots, v_k và đi qua các cạnh e_1, e_2, \dots, e_l . Khi đó, ta định nghĩa trọng số của đường đi này là $\max(w(v_1), w(v_2), \dots, w(v_k)) * \max(c(e_1), c(e_2), \dots, c(e_l))$. Nói cách khác, trọng số của một đường đi là tích của **trọng số lớn nhất của một đỉnh trên đường đi** (bao gồm đỉnh xuất phát và đỉnh kết thúc) và **trọng số lớn nhất của một cạnh trên đường đi**.

Với hai đỉnh u và v trên đồ thị, gọi $d(u, v)$ là trọng số nhỏ nhất của một đường đi từ u đến v . Nếu $u = v$, ta có $d(u, v) = 0$. Với mỗi đỉnh u của đồ thị, hãy tính $S(u) = d(u, 1) + d(u, 2) + \dots + d(u, n)$.

INPUT

Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 5$) là số thứ tự của subtask chứa test này.

Dòng thứ hai chứa hai số nguyên n và m ($1 \leq n \leq 500$; $n - 1 \leq m \leq n * (n - 1) / 2$) lần lượt là số đỉnh và số cạnh của đồ thị.

Dòng thứ ba chứa n số nguyên w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^7$) lần lượt là trọng số của các đỉnh.

m dòng cuối cùng, dòng thứ j chứa ba số nguyên f_j, t_j và c_j ($1 \leq f_j, t_j \leq n, 1 \leq c_j \leq 10^7$) cho biết có một cạnh của đồ thị nối hai đỉnh f_j, t_j và có trọng số là c_j . Dữ liệu vào đảm bảo đồ thị này liên thông.

OUTPUT

Một dòng duy nhất với n số nguyên $S(1), S(2), \dots, S(n)$.

GIỚI HẠN

Subtask 1 (14% số điểm): Tất cả các đỉnh và tất cả các cạnh đều có trọng số bằng 1.

Subtask 2 (20% số điểm): Tất cả các đỉnh có trọng số bằng 1.

Subtask 3 (20% số điểm): Tất cả các cạnh có trọng số bằng 1.

Subtask 4 (20% số điểm): $n \leq 50$

Subtask 5 (26% số điểm): Không có ràng buộc gì thêm.

nmgraph.inp	nmgraph.out
4	264 174 174
3 3	
5 6 3	
1 2 22	
2 3 7	
3 1 97	

Giải thích: Trong ví dụ trên, ta có:

- $d(1, 2) = d(2, 1) = 22 * 6 = 132$
- $d(2, 3) = d(3, 2) = 7 * 6 = 42$
- $d(3, 1) = d(1, 3) = 22 * 6 = 132$

TRANS

Cho một hoán vị p (chỉ số bắt đầu từ 0) của n số nguyên dương đầu tiên. Lúc đầu, $p_i = i$ ($0 \leq i < n$). Bạn cần lần lượt thực hiện q thao tác, mỗi thao tác có một trong hai dạng sau:

- **R l r** ($0 \leq L \leq R < n$): Đảo ngược đoạn con liên tiếp từ chỉ số L đến chỉ số r của p ($p_L, p_{L+1}, \dots, p_r \rightarrow p_r, p_{r+1}, \dots, p_L$)
- **S l r a** ($0 \leq l \leq a \leq r < n$): Dịch đoạn con liên tiếp từ chỉ số l đến chỉ số r của p theo vòng tròn sao cho p_l sau khi thực hiện thao tác bằng p_a trước khi thực hiện thao tác ($p_l, p_{l+1} \dots, p_r \rightarrow p_a, p_{a+1}, \dots, p_r, p_l, \dots, p_{a-1}$).

Sau mỗi thao tác, hãy tìm số cặp nghịch thế (số cặp chỉ số (i, j) thỏa mãn $0 \leq i < j < n$ và $p_i > p_j$).

INPUT

Dòng đầu tiên chứa hai số nguyên n ($1 \leq n \leq 1,000,000$) và q ($1 \leq q \leq 1,000,000$). q dòng tiếp theo, mỗi dòng là một trong hai dạng thao tác đã được miêu tả ở trên.

OUTPUT

Sau mỗi thao tác, hãy in ra trên một dòng riêng biệt số cặp nghịch thế của p hiện tại modulo 2.

GIỚI HẠN

- 25% số điểm: $n \leq 1,000, q \leq 1,000$.
- 25% số điểm : $n \leq 100,000, q \leq 100,000$.
- 50% số điểm : Không có ràng buộc gì thêm.

trans.inp	trans.out
2 3	0
S 1 1 1	1
R 0 1	1
R 0 0	
2 3	1
R 0 1	1
S 0 1 0	1
R 1 1	

SCI

Cho số N , thầy Hùng muốn chia số N thành tổng của các số nguyên dương liên tiếp. Ví dụ, với $N=15$ ta có 3 cách chia:

- $15=1+2+3+4+5$
- $15=4+5+6$
- $15=7+8$

Cho số N , hãy đếm số cách chia số N thành tổng các số nguyên dương liên tiếp.

INPUT

Dòng 1: số N

OUTPUT

Dòng 1: số cách chia

GIỚI HẠN

- Subtask 1 (16% số test) $N < 10^2$
- Subtask 2 (24% số test) $N < 10^6$
- Subtask 3 (36% số test) $N < 10^{10}$
- Subtask 4 (24% số test) $N < 10^{16}$

sci.inp	sci.out
15	3