

# 2013-2014. Thanh Hoa.

## Ngày 1

### A1507

Dù học với nhau nhiều, nhưng Dự tuyển vẫn chưa thân. Phễu đại ca hy vọng tổ chức một cuộc đi chơi, lôi lũ đàn em ra khỏi máy tính, tham gia các hoạt động giải trí, qua đó nhiều bé chưa biết nhau sẽ biết nhau. Các bé tham gia các hoạt động chưa biết nhau càng tốt.

Tổ chức cuộc đi chơi thành công là việc không dễ dàng, không chỉ cần có học sinh tham gia, mà còn cần đội tình nguyện. Đội tình nguyện sẽ mất rất nhiều công sức để chuẩn bị, vì vậy thành viên của đội tình nguyện đều cần phải biết nhau.

Dự tuyển có  $N$  học sinh, đánh số từ 1 đến  $N$ . Có  $M$  mối quen biết giữa các cặp học sinh, và phần còn lại không biết nhau.

Phễu hy vọng chọn một số học sinh để thành lập một nhóm tình nguyện chịu trách nhiệm về tổ chức các hoạt động, và tất cả những người không phải tình nguyện viên sẽ trở thành người tham gia hoạt động. Phễu yêu cầu:

1. Thành viên của nhóm tình nguyện phải biết nhau.
2. Sinh viên tham gia vào các hoạt động không được biết nhau;
3. Ít nhất một bạn tham gia nhóm tình nguyện, và ít nhất một bạn tham gia hoạt động.

Phễu muốn biết có bao nhiêu cách tổ chức nhóm tình nguyện ?

### INPUT

Dòng đầu tiên ghi số test  $T$ , sau đó là  $T$  bộ test theo định dạng sau:

Dòng đầu tiên ghi 2 số nguyên  $N$  và  $M$ .

$M$  dòng tiếp theo, dòng thứ  $i$  ghi 2 số nguyên  $x_i, y_i$ , cho biết 2 học sinh  $x_i$  và  $y_i$  quen nhau.

Đảm bảo không có mối quan hệ bạn bè nào giữa 2 học sinh được liệt kê 2 lần.

### OUTPUT

Đối với mỗi test, in ra tổng số giải pháp khả thi theo mod 1000003.

Sample Input	Sample Output
2	0
1 0	3
4 4	
1 2	
1 3	
2 3	
3 4	

## GIỚI HẠN

- $N \leq 10$  cho 20% số test;
- $N \leq 30$  cho 40% số test;
- $1 \leq N \leq 1000, 0 \leq M \leq N^2, 1 \leq T \leq 6$ .

## A1508

Đây là đoạn code C ++ cài đặt thuật toán sắp xếp nổi bọt:

```
for (int i = 1; i < n; i++)
    for (int j = 1; j ≤ n-i; j++)
        if (a [j] > a [j + 1])
            swap (a [j], a [j + 1]);
```

Mảng a có n phần tử đánh số từ 1 đến n cần được sắp xếp lại, và hàm swap đổi vị trí 2 số liên kề.

Cho mảng a và số k, mảng sẽ như thế nào sau đúng k lần swap bằng đoạn code trên?

## INPUT

Dòng 1 ghi 2 số n và k.

Dòng 2 ghi n số nguyên dương phân biệt của mảng a.

## OUTPUT

2

In ra Impossible nếu không thể. Ngược lại, in ra mảng a.

Sample Input	Sample Output
1 1 1	Impossible!
5 5 5 4 3 2 1	3 4 2 1 5

Test	$n \leq$	$k \leq$
1	10	$10^{12}$
2	5000	$10^{12}$
3	20000	$10^{12}$
4	$10^6$	$10^6$
5	$10^6$	$2 \cdot 10^6$
6	$10^6$	$2 \cdot 10^6$
7	$10^5$	$10^{12}$
8	$10^5$	$10^{12}$
9	$10^6$	$10^{12}$
10	$10^6$	$10^1$

# A1509

Phiếu có N thẻ, mỗi thẻ có ghi 1 số (từ 0 đến 9) và tô bởi một trong 3 màu: đỏ (0), vàng (1), xanh (2). Phiếu muốn đặt N thẻ liên tiếp nhau, sao cho 2 thẻ liền kề hoặc cùng màu hoặc cùng số. Hỏi Phiếu có bao nhiêu cách làm điều đó.

## INPUT

Dòng đầu tiên ghi số thẻ N.

Mỗi dòng trong N dòng tiếp theo ghi màu sắc và số ghi trên mỗi thẻ.

## OUTPUT

3

In ra đáp số theo modulo  $2^{32}$ .

Sample Input	Sample Output
3 0 0 0 0 0 0	6
3 0 1 0 2 1 2	2

## GIỚI HẠN

- 10% dữ liệu:  $n \leq 10$
- Thêm 10% dữ liệu:  $n \leq 20$
- 30% dữ liệu khác:  $n \leq 30$ , chỉ có 2 màu, và không có 3 lá bài có cùng số và màu sắc.
- 50% dữ liệu khác:  $n \leq 30$ , chỉ có 3 màu, và không có 4 thẻ có cùng số và màu sắc.

# 2013-2014. Thanh Hoa. Ngày 2

## A1510

**Tôi chưa bao giờ nói "640K là đủ cho bất cứ ai!"**

**Bill Gates**

Người bạn tốt Bánh Ngọt của Phễu thường inbox với Phễu. Tuy nhiên, hai bạn rất sợ Tò Mò đọc trộm và mang tin tức ra share. Họ đã đưa ra một cách mã hóa để bảo vệ tính nhắn. Dù rất an toàn, nhưng phương pháp này cần 2 bên thỏa thuận trước một chuỗi ngẫu nhiên. Tuy nhiên Bánh Ngọt không thể nhớ một chuỗi dài. Để giúp đỡ, Phễu đề xuất một thuật toán "nhân ngẫu nhiên". Thuật toán này

biến một chuỗi N-bit ngẫu nhiên thành một chuỗi  $2*N$  bit tựa ngẫu nhiên. Cụ thể, với chuỗi N bit, chọn 2 tham số M, C, và sinh ra chuỗi  $2*N$  bit theo cách sau:

Đầu tiên đặt  $H = \text{Mod}(1371, 2^M)$ ,  $s = 0$

Với mỗi i chạy từ 1 đến N, nếu bit thứ i ở đầu vào là  $X_i$ , thực hiện:

$S = \text{Mod}(\text{RotL}(\text{Mod}(3432918353 * s + C, 2^{32}), 32) * 461845907, 2^{32})$

$H = \text{Xor}(H, \text{Xor}(\text{Mod}(s, 2^M), X_i))$

$H = \text{RotL}(H, M)$

$H = \text{Mod}(H * 5 + 3864292196, 2^M)$

M-1 bit (ví dụ, 1 và 1 nếu  $H = 13$  (nhị phân là 1101),  $M = 4$ )

Ở đây, Mod là phép toán lấy dư khi lấy a chia cho b.

RotL(a, b) là dịch sang trái b bit, nghĩa là, tất cả tất cả các bit của a di chuyển đến một vị trí cao hơn, vị trí ban đầu của b di chuyển trở thành bit đầu tiên. (Ví dụ,  $\text{RotL}(13, 4) = 11$ , bởi vì  $13_2 = 1101$ , xoay xong trở thành  $1011 = 11_{10}$ )

Từ các hằng số trong thuật toán này, chúng ta có thể thấy rằng thuật toán này được chuyển thể từ thuật toán MurmurHash3 nổi tiếng.

Tò Mò cho rằng thuật toán này chưa ngẫu nhiên lắm. Chúng ta biết rằng nếu thuật toán này là thực sự ngẫu nhiên, bởi vì đầu vào N bit, nên không có quá nhiều 0 trong đầu ra. Nhưng Phieu muốn sinh ra 1 chuỗi đầu vào, sao cho chuỗi đầu ra có nhiều 0 nhất.

## INPUT

3 số N, M, C

## OUTPUT

In ra số số 1 tối thiểu trong chuỗi đầu ra.

Dòng thứ 2 ghi N bit 01 là chuỗi đầu vào của thuật toán.

Sample Input	Sample Output
3 3 5	0

## GIỚI HẠN

- Với 30% số test,  $N \leq 20$ ,  $M \leq 5$
- Đối với 50% số test,  $N \leq 1000$ ,  $M \leq 6$
- $1 \leq N \leq 5000$ ,  $3 \leq M \leq 20$ ,  $N * 2^M \leq 50000000$ ,  $0 \leq C < 2^{31}$

## A1511

Phễu muốn đi tìm bạn gái, nhưng những người bạn gái không ở cùng một thành phố. Phễu muốn biết: Có bao nhiêu lựa chọn đến thành phố có bạn bè nữ?

Phễu ở một nơi có  $n$  thành phố, đánh số từ 1 đến  $n$ . Bất kỳ hai thành phố nào đều được kết nối bằng những con đường một chiều. Mỗi thành phố  $u$  có  $k$  điểm vào:  $in[u][1]$ ,  $in[u][2]$  ...  $in[u][k]$ , và  $k$  điểm ra:  $ou[u][1]$ ,  $ou[u][2]$  ...  $ou[u][k]$ . Đối với bất kỳ hai thành phố  $(u, v)$  ( $u$  có thể là  $v$ ), số lượng con đường từ  $u$  đến  $v$  là  $s(ou[u][1] * in[v][1] + ou[u][2] * in[v][2] + ... + ou[u][k] * in[v][k])$ .

Phễu được hỏi  $m$  truy vấn dạng  $(u, v, d)$ , yêu cầu đếm xem có bao nhiêu cách đi từ thành phố  $u$  đến thành phố  $v$  bằng không quá  $d$  con đường.

## INPUT

Dòng đầu tiên ghi 2 số nguyên dương  $n, k$ .

Mỗi dòng trong  $n$  dòng tiếp theo ghi  $2k$  số nguyên, ở dòng thứ  $i$  ghi  $2k$  số:  $ou[i][1], ou[i][2], ..., ou[i][k], in[i][1], in[i][2], ..., in[i][k]$

Dòng tiếp theo ghi số truy vấn  $m$ .

Mỗi dòng trong  $m$  dòng tiếp theo ghi 3 số nguyên:  $u, v, d$ .

$1 \leq u, v \leq n$ . Mọi số nguyên đọc vào  $\leq 2147483647$ .

## OUTPUT

Đối với mỗi truy vấn, in ra một số theo modulo 1000000007.

Sample Input	Sample Output
5 2	1
2 5 4 3	51
7 9 2 4	170107227
0 1 5 2	271772358
6 3 9 2	34562176
2147483647 1000000001	890241289
233522 788488	8516097
10	383966304
1 1 0	432287042
2 2 1	326522835
2 4 5	
4 3 10	
3 4 50	
1 5 1000	
3 5 1000000000	
1 2 500000000	
4 5 2147483647	
3 1 2147483647	

Số test	n	k	m
1-3	20	5	19
4	1000	1	45
5-7	50	20	45
8-10	1000	20	50

//A1512

**Chúa Trời luôn luôn là nhà hình học.  
- Plato -**

Trong mơ, Phêru đến một xứ sở thịnh vượng, giao thông phát triển.

Các thành phố của xứ sở lý tưởng được đánh số khác nhau từ 1 đến n. Mỗi thành phố coi như 1 điểm trên mặt phẳng tọa độ. Tổng cộng có m con đường cao tốc nối các cặp thành phố. Các con đường cao

tốc chỉ giao nhau ở các thành phố đầu mút. Các thành phố được kết nối kép, nghĩa là giữa bất kỳ 2 thành phố nào, cũng có hai tuyến đường đơn giản kết nối chúng và 2 tuyến đường này không giao nhau.

Mỗi thành phố có một mức độ thịnh vượng khác nhau, có giá trị từ 1 (giàu nhất), đến  $n$  (nghèo nhất). Nếu độ thịnh vượng của 2 thành phố đầu mút của một con đường là  $a$  và  $b$  ( $a < b$ ), thì khoảng mở  $(a, b)$  là chỉ số cuốn hút của con đường. Định nghĩa 2 con đường tương tự nhau khi và chỉ khi giao điểm giữa chỉ số cuốn hút của chúng không rỗng.

Phần thức dậy, chỉ nhớ mạng giao thông của các thành phố ở xứ sở lý tưởng, và quên đi mức độ thịnh vượng của mỗi thành phố. Cậu muốn trả lời 2 câu hỏi sau.

- (1) Trung bình có bao nhiêu con đường tương tự nhau?
- (2) Trong tình huống tốt nhất, tối thiểu có bao nhiêu con đường tương tự nhau?

## INPUT

Dòng đầu tiên ghi 2 số nguyên  $n, m$ , là số thành phố và số con đường cao tốc.

Mỗi dòng trong  $n$  dòng tiếp theo, ghi hai số nguyên, là tọa độ của thành phố có nhãn  $i$  trên bản đồ.

Mỗi dòng trong  $m$  dòng tiếp theo ghi 2 số nguyên, là số thứ tự 2 thành phố đầu mút của con đường.

## OUTPUT

In ra kết quả 2 câu hỏi. Câu trả lời cho câu hỏi đầu tiên phải có 3 chữ số sau dấu chấm thập phân.

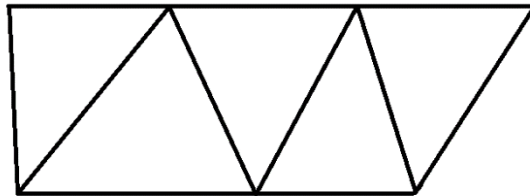
Sample Input	Sample Output
3 1 9 2 8 0 0	2.000 2



1 2	
2 3	
3 1	

## GIỚI HẠN

- 10% số test,  $n \leq 9$ .
- 20% số test,  $n \leq 13$ .
- 30% số test khác, bố cục đơn giản là đa giác, biểu đồ lưới, hình tam giác.
- 10% số test, bao gồm không quá 10 dòng, mỗi dòng gồm toàn hình vuông hoặc hình tam giác. Dòng hình vuông có không quá 10 hình vuông, dòng tam giác có không quá 20 tam giác.
- 10% số test còn lại chứa không quá 13 mặt.
- Đối với 100% số test,  $n \leq 100$ , tọa độ tuyệt đối nằm trong phạm vi 30.000. Chúng tôi đồng ý rằng hình dạng của tam giác có hình giống như hình dưới đây. Một trong những cạnh song song của nó song song với trục tọa độ. Không chỉ vậy, cạnh của lưới đồ thị song song với các trục



# 2013-2014. Thanh Hoa.

## Ngày 3

### A1513

Có một mảng  $n$  số  $\{a_1, a_2, \dots, a_n\}$ .  $m$  truy vấn, tìm số tự nhiên nhỏ nhất không xuất hiện trong một khoảng.

#### INPUT

Dòng đầu tiên ghi 2 số  $n, m$ .

Dòng thứ hai là  $n$  số của mảng.

Bắt đầu từ dòng thứ ba, mỗi dòng ghi một truy vấn  $[l, r]$ .

#### OUTPUT

Với mỗi truy vấn, in trên một dòng câu trả lời.

Sample Input	Sample Output
5 5	1
2 1 0 2 1	2
3 3	3
2 3	0
2 4	3
1 2	
3 5	

#### GIỚI HẠN

- Đối với 100% số test:  $1 \leq n, m \leq 200000, 0 \leq a_i \leq 10^9, 1 \leq l \leq r \leq n$
- Đối với 30% số test:  $1 \leq n, m \leq 1000$

//A1514

Sample Input	Sample Output

## A1515

Một bộ sinh chuỗi tạo ra chuỗi ký tự, ban đầu tạo ra một chuỗi rỗng, mỗi lần, tạo ra một chữ thường với một xác suất nào đó, đặt vào sau chuỗi hiện thời.

Cho  $N$  chuỗi có chiều dài  $L$ , mỗi chuỗi chỉ có các ký tự chữ thường.

Nếu lúc nào đó, phát hiện ra có một chuỗi trong  $N$  chuỗi là chuỗi con trong chuỗi được tạo ra ở lúc đó, thì bộ sinh chuỗi sẽ ngừng hoạt động và in ra chuỗi hiện thời.

Tính độ dài kỳ vọng chuỗi được sinh ra.

### INPUT

Dòng đầu tiên ghi số test  $C$ . Sau đó là các test có dạng sau:

Dòng đầu tiên của mỗi bộ số test ghi 3 số nguyên dương  $N, L, T$  ( $L \leq 6, T \leq 10$ ).  $T$  có nghĩa là bộ sinh sẽ chỉ tạo ra các chữ thường trong  $T$  chữ cái đầu tiên.

Mỗi dòng từ dòng số 2 đến  $N+1$ , ghi một chuỗi chiều dài  $L$ , mỗi chuỗi bao gồm các chữ thường (thuộc  $T$  chữ cái đầu tiên).

Dòng  $N+2$  ghi  $T$  số nguyên dương không quá 10.000, số thứ  $i$  là số nguyên  $x_i$ , nghĩa là xác suất tạo ra chữ cái thứ  $i$  là  $x_i/10000$ . Đảm bảo tổng của  $T$  số nguyên dương là 10000.

### OUTPUT

Với mỗi test, in ra trên một dòng giá trị kỳ vọng độ dài chuỗi đầu ra.

Sample Input	Sample Output
1 2 3 3 aac abb 3333 3333 3334	40.5060771264

## GIỚI HẠN

- Với 30% số test,  $N = 1$ ,  $C = 1$ ;
- Đối với 60% số test,  $N \leq 4$ ,  $C \leq 2$ ;
- $N \leq 8$ ,  $C \leq 20$  cho 100% số test.

# 2013-2014. Thanh Hoa.

## Ngày 4

### A1516

Đối với một thập phân  $n$ -bit  $x$  ( $A_n A_{n-1} \dots A_1$ ), chúng ta định nghĩa trọng số của nó như sau:

$$F(x) = A_n * 2^{n-1} + A_{n-1} * 2^{n-2} + \dots + A_1 * 2^0$$

Bây giờ, bạn có 2 số thập phân  $A$  và  $B$ , đếm xem trong khoảng đóng  $[0, B]$  có bao nhiêu số  $x$  có trọng số không lớn hơn trọng số của  $A$ , tức là  $F(x) \leq F(A)$ . In ra kết quả theo mod  $(10^9 + 7)$ .

## INPUT

Dòng đầu tiên ghi số test  $T$ .

Sau đó  $T$  dòng, mỗi dòng mô tả một test, gồm 2 số nguyên không âm  $A, B$ .

## OUTPUT

Đối với mỗi test, in ra trên một dòng "Case #t: ans", trong đó ans biểu thị đáp án.

Sample Input	Sample Output
3	Case #1: 1
0 100	Case #2: 2
1 10	Case #3: 13
5 100	

Đối với test #3, có 13 số 0, 1, 2, 3, 4, 5, 10, 11, 12, 13, 20, 21, 100, thỏa mãn.

## GIỚI HẠN

- Đối với 20% số test,  $A, B \leq 10^9$ ;
- Đối với 30% số test,  $A, B \leq 10^{18}$ ;
- Đối với 100% số test,  $A, B \leq 10^{200}$ ,  $T \leq 5$ .

## A1517

Phễu trồng một cây động, một số nút sẽ mọc trái cây vào mỗi ngày. Cây có n nút, n-1 cạnh. Nút 1 là gốc.

Đừng quên rằng đây là một cây động, linh hoạt trong từng khoảng khác. Phễu yêu cầu bạn duy trì 2 sự kiện trên cây.

- Sự kiện 0: Trái cây mọc. Mỗi nút trên một cây con sẽ mọc thêm K quả.
- Sự kiện 1: Phễu yêu cầu bạn đếm số trái cây trên một vài nhánh. Nhánh là một phần trên một đường đi từ một nút đến nút gốc. Mỗi lần Phễu sẽ lựa chọn một vài nhánh, và bạn phải đếm số quả trên các nút nằm trên các nhánh đó. Lưu ý các nhánh có thể chồng lên nhau, nhưng các nút chỉ được tính một lần.

Ban đầu, các nút không có quả.

## INPUT

Dòng đầu tiên ghi số nút n ( $1 \leq n \leq 200.000$ ).

Mỗi dòng trong  $n-1$  dòng tiếp theo ghi 2 số  $u, v$  thể hiện có cạnh nối nút  $u$  và nút  $v$ .

Dòng kế tiếp ghi số sự kiện  $NQ$  ( $1 \leq NQ \leq 200.000$ ).

Mỗi dòng trong  $NQ$  dòng tiếp theo mô tả một sự kiện. Số đầu tiên trong mỗi dòng là kiểu sự kiện (0 hoặc 1).

Nếu là sự kiện 0, tiếp sau là 2 số nguyên  $u, \delta$ , với ý nghĩa mỗi nút trong cây con có gốc là nút  $u$  sẽ mọc thêm  $\delta$  trái cây.

Nếu là sự kiện 1, tiếp theo là số nguyên  $K$  ( $1 \leq K \leq 5$ ), với ý nghĩa phải tính toán  $K$  nhánh. Sau đó,  $K$  cặp số nguyên  $u_k, v_k$ , thể hiện một nhánh đi từ  $u_k$  đến  $v_k$ . Vì số trái cây có thể rất lớn, in ra theo mod  $2^{31}$ .

## OUTPUT

Đối với mỗi sự kiện 1, in ra câu trả lời.

Sample Input	Sample Output
5 1 2 2 3 2 4 1 5 3 0 1 1 0 2 3 1 2 3 1 1 4	13

## GIỚI HẠN

- Test 1 :  $1 \leq n \leq 2.000, 1 \leq NQ \leq 2.000, K \leq 5$
- Test {2, 3} :  $1 \leq n \leq 100.000, 1 \leq NQ \leq 100.000, K = 1$
- Test {4, 5} :  $1 \leq n \leq 100.000, 1 \leq NQ \leq 100.000, K \leq 2$
- Test {6, 7} :  $1 \leq n \leq 100.000, 1 \leq NQ \leq 100.000, K \leq 3$
- Test {8, 9, 10} :  $1 \leq n \leq 200.000, 1 \leq NQ \leq 200.000, K = 5$ .

# A1518

Phiếu gần đây chơi trò khai thác khí.

Trên bản đồ có  $n$  điểm, được nối bởi  $m$  cạnh một chiều. Để thuận tiện, đánh số các điểm bằng một số nguyên khác nhau từ 1 đến  $n$ .

Phiếu có vô hạn khí. Vào đầu game, tất cả lỗ khí đều ở điểm 1. Mỗi vòng, tất cả khí phải di chuyển đến điểm tiếp theo dọc theo một cạnh trực tiếp trong bản đồ. Người chơi có thể chỉ định các tuyến khác nhau cho các con khí khác nhau, nhưng khí không thể đứng yên. Con khí rời đi sẽ biến mất trong vòng tiếp theo.

Trò chơi bắt đầu với lượt 0. Từ  $S$  vòng đầu tiên, cho đến khi kết thúc vòng  $T$  (kể cả vòng vòng  $S$  và  $T$ ), ở mỗi vòng, mỗi điểm sẽ tạo ra 1 đơn vị quặng. Nếu ít nhất một con khí ở điểm đó ở vòng này, người chơi sẽ nhận được quặng, ngược lại quặng sẽ biến mất trong vòng tiếp theo.

Bây giờ, Phiếu muốn biết có bao nhiêu quặng có thể nhận được tại mỗi điểm trong bản đồ.

## INPUT

Dòng đầu tiên ghi 4 số nguyên dương  $n, m, S, T$ , tương ứng, số các điểm trên bản đồ, số cạnh, thời điểm bắt đầu và kết thúc việc tạo quặng.

Sau đó  $m$  dòng, mỗi dòng ghi 2 số nguyên  $x, y$ , thể hiện có 1 cạnh từ điểm  $x$  đến điểm  $y$ .

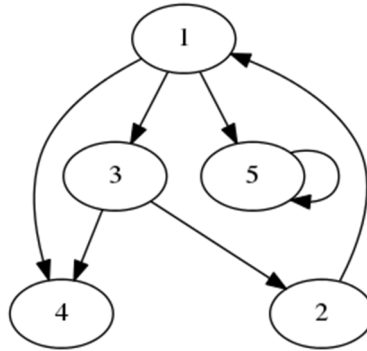
## OUTPUT

In ra  $n$  dòng, dòng thứ  $i$  là số quặng kiếm được từ điểm  $i$ .

Sample Input	Sample Output
5 7 0 19	7
1 3	6
1 4	7
1 5	13
3 2	19
2 1	

3 4	
5 5	

Ví dụ về đồ thị ở hình bên.  
 Vòng lặp  $1 \rightarrow 3 \rightarrow 2$  cho phép lũ khi đi vòng vòng, quay về nút 1 sau 3 vòng, lấy được 1, 2, 3 đơn vị quặng. Ngoài ra, lũ khi đi theo  $1 \rightarrow 4$  và  $1 \rightarrow 3 \rightarrow 4$  cũng kiếm được quặng. Sau vòng 1, có 1 con khi đến 5, và nó có thể đi vòng quanh ở đó để kiếm quặng



Số quặng kiếm được ở mỗi điểm như sau:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	o	x	x	o	x	x	o	x	x	o	x	x	o	x	x	o	x	x	o	x
2	x	x	o	x	x	o	x	x	o	x	x	o	x	x	o	x	x	o	x	x
3	x	o	x	x	o	x	x	o	x	x	o	x	x	o	x	x	o	x	x	o
4	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	x	o
5	x	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o

## GIỚI HẠN

- 10% số test: với hai điểm  $x \neq y$  không có đường đi từ  $y$  đến  $x$  nếu có đường đi từ  $x$  đến  $y$ .
- 40% khác của số test có  $S \geq 9 \times 10^{14}$ . Trong số đó, có 20% tổng số số test có  $T-S \leq 50000$ .
- Đối với 100% số test,  $0 < n \leq 100$ ,  $0 < m \leq 300$ ,  $0 \leq S \leq T \leq 10^{15}$ .