Shape, square

Description automatically generated

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG: ĐH GIAO THÔNG VẬN TẢI**

**KHOA: CNTT**

-----🙞🙜🕮🙞🙜-----



**BÀI TẬP LỚN**

*Môn Học : Thuật toán và ứng dụng*

*Đề tài: Ngắm sao*

Giáo viên hướng dẫn: Phạm Xuân Tích

Sinh viên thực hiện: Nguyễn Thị Ngũ \_ 191200758

Lớp: Công nghệ thông tin 4

Khóa: 60

Hà Nội, năm 2021

Mục lục

[**1.** **Nêu bài toán** 3](#_Toc87741425)

[**2.** **Nêu ý tưởng** 3](#_Toc87741426)

[**3.** **Mô phỏng tính tính toán bằng tay** 4](#_Toc87741427)

[**4.** **Code C/C++ hoặc python** 5](#_Toc87741428)

[**5.** **Đánh giá độ phức tạp** 6](#_Toc87741429)

1. **Nêu bài toán**

MAH có sở thích ngắm sao. Hôm nay anh ấy thấy n ngôi sao trên bầu trời, ngôi sao thứ i có tọa độ (xi, yi) và độ sáng tối đa là c, và độ sáng ban đầu của các ngôi sao là si(0 <= si <= c).

Ở thời điểm ban đầu (t = 0), ngôi sao thứ có độ sáng si. Nếu ở thời điểm t một ngôi sao có độ sáng là x thì thời điểm (t + 1) ngôi sao này có độ sáng là x + 1 khi x + 1 <= c và có độ sáng 0 trong trường hợp còn lại.

MAH muốn nhìn bầu trời q lần. Trong lần thứ i anh ấy sẽ nhìn vào thời điểm ti và anh ấy sẽ nhìn một hình chữ nhật có các cạnh song song với trục tọa độ, góc trên bên trái là (x1i,y1i) và góc dưới bên phải là (x2i,y2i). Mỗi lần nhìn, anh ấy muốn biết tổng độ sáng của tất các các ngôi sao nằm trong hình chữ nhật (có thể nằm trên đường viền).

Input:

Dòng đầu gồm 3 số nguyên n, q, c (1 ≤ n, q ≤ 10^5, 1 ≤ c ≤ 10) biểu diễn số ngôi sao, số lần nhìn, và độ sáng tối đa của các ngôi sao.

N dòng tiếp theo biểu diễn các ngôi sao. Dòng thứ i chứa 3 số nguyên xi, yi, si (1 ≤ xi, yi ≤ 100, 0 ≤ si ≤ c ≤ 10) biểu diễn tọa độ của ngôi sao thứ i và độ sáng ban đầu.

Q dòng tiếp theo mỗi dòng chữa 5 số nguyên ti, x1i, y1i, x2i, y2i (0 ≤ ti ≤ 109, 1≤ x1i < x2i ≤ 100, 1≤ y1i < y2i ≤ 100) biểu diễn thời điểm nhìn và tọa độ của hình chữ nhật.

Output:

Mỗi dòng in tổng độ sáng trong mỗi lần nhìn.

1. **Nêu ý tưởng**

- Trường hợp test yếu: tìm các ngôi sao nằm trong hình chữ nhật. Áp dụng công thức tính độ sáng của ngôi sao có tọa độ xi, yi.

=> Do dùng 2 vòng lặp nên thời gian chạy sẽ lâu nên ta dùng quy hoạch động cho bài này.

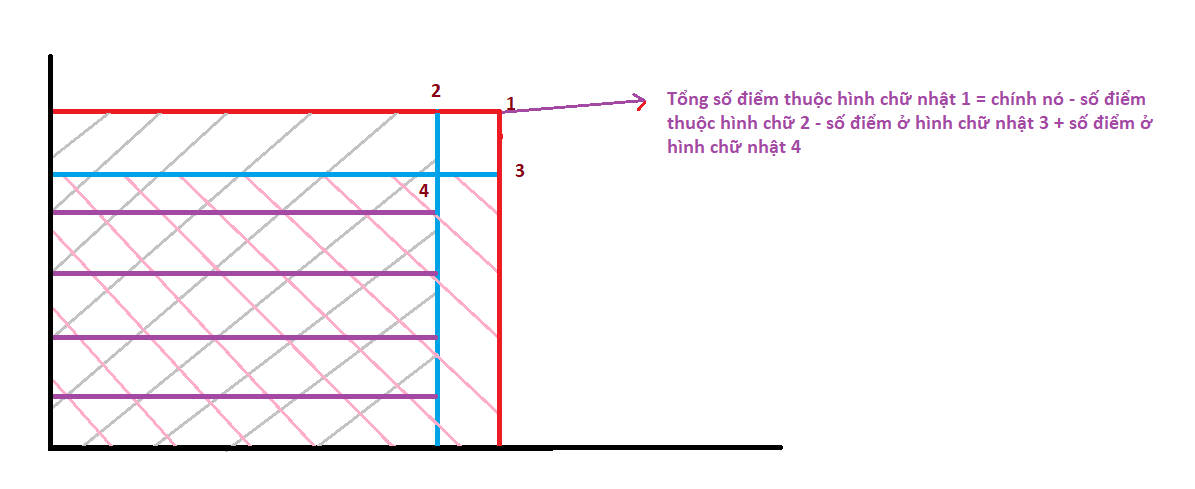
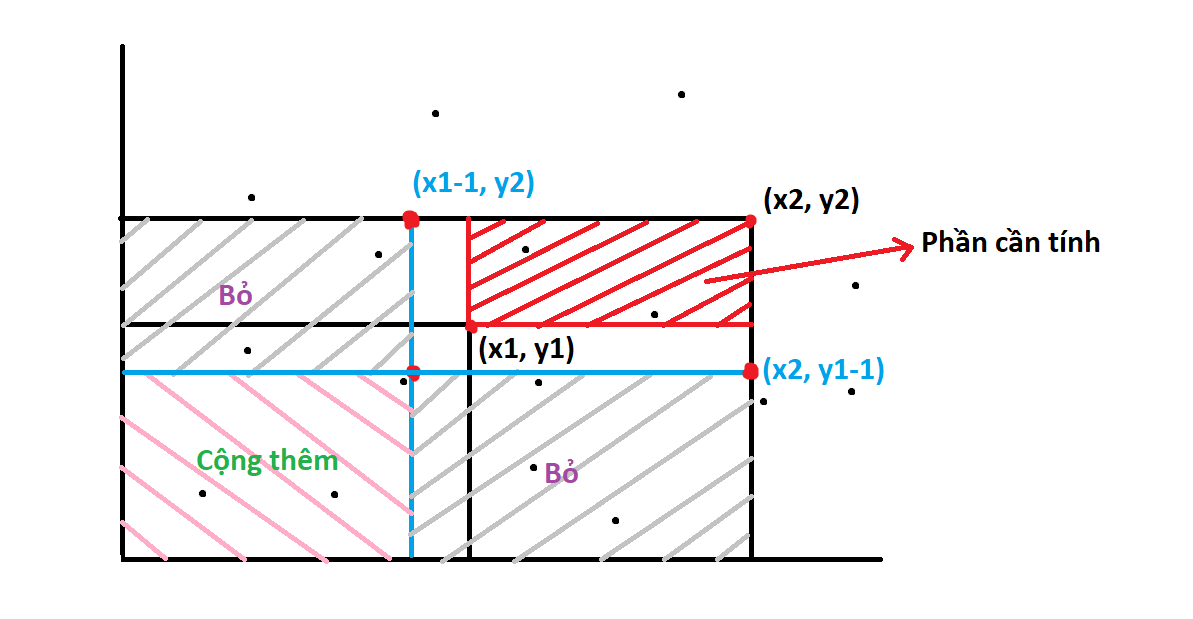
Ý tưởng:

- Nhập tọa độ và độ sáng ban đầu của các ngôi sao. Đếm số lượng ngôi sao ở tọa độ xi, yi và có độ sáng si.

- Gọi P[ti][xi][yi] là số ngôi sao có độ sáng ban đầu là t thuộc hình chữ nhật có 2 đỉnh (0,0) và (xi, yi). Ta tìm tất cả P[ti][xi][yi] với 1 ≤ ti ≤ c, 1≤ xi < xi ≤ 100

- Chạy bộ test: Với mỗi hình chữ nhật, số ngôi sao ở hình chữ nhật có 2 đỉnh (x1i, y1i) và (x2i, y2i) = tổng của ( P[ti][x2i][y2i] - P[ti][xi -1][yi] - P[ti][xi][yi -1] + P[ti][xi -1][yi-1]) nhân với độ sáng tại thời điểm test.

1. **Mô phỏng tính tính toán bằng tay**

****

VD:

2 1 5

40 6 4

60 48 4

17 4 1 47 38

- P[4][40][6] = 1; // thời điểm ban đầu t = 4 tọa độ (40, 6) có 1 điểm

- P[4][60][48] = 1; // thời điểm ban đầu t = 4 tọa độ (60, 48) có 1 điểm

=> Quy hoạch động giá trị của P[1][1][1] đến P[5][100][100]:

- t = 1:

+P[1][1][1] = P[1][1][1] - P[1][0][1]- P[1][1][0] + P[1][0][0]; // tọa độ (1, 1)

+P[1][1][2] = P[1][1][2] - P[1][0][2]- P[1][1][1] + P[1][0][1]; // tọa độ (1, 2)

..... tương tự

- t = 4:

+P[4][1][1] = P[4][1][1] - P[4][0][1]- P[4][1][0] + P[4][0][0]; // tọa độ (1, 1)

....

+P[4][40][6] = P[4][40][6] - P[4][39][6]- P[4][40][5] + P[4][39][5]; // tọa độ (1, 2)

= 1 – 0 – 0 + 0 = 1

..... tương tự

- t = 5:

+P[5][1][1] = P[5][1][1] - P[5][0][1]- P[5][1][0] + P[5][0][0]; // tọa độ (1, 1)

+P[5][1][2] = P[5][1][2] - P[5][0][2]- P[5][1][1] + P[5][0][1]; // tọa độ (1, 2)

.....

\* Các test: hình chữ nhật (4,1), (47, 38)=> Chỉ có điểm (6,4) nằm trong hình chữ nhật

- t = 1: P[1][][] = 0

- t = 2: P[2][][] = 0

- t = 3: P[3][][] = 0

- t = 4: P[4][][] = 1

=> KQ = 0\*(1+4%(5+1))%(4+1) + 0\*(2+4%(5+1))%(4+1) + 0\*(3+4%(5+1))%(4+1) + 1\*(4+17%(5+1))%(4+1) = 0 + 0+ 0+ 3 = 3

1. **Code C/C++ hoặc python**

#include<bits/stdc++.h>

#define ll long long

#define x first

#define y second

using namespace std;

int main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(0); cout.tie(0);

int n, q, c, t, x1, y1, x2, y2, d;

cin>>n>>q>>c;

int p[15][102][102];

int xi, yi, si;

for(int i=0;i<n;i++){

cin>>xi>>yi>>si;

p[si][xi][yi]++;

}

for(int t=1; t<=c; t++) for(int i=1; i<=100; i++) for(int j=1; j<=100; j++){ //dem so luong diem co do sang ban dau la t tai hcn co dinh (0,0) va (i,j)

p[t][i][j] += p[t][i-1][j] + p[t][i][j-1] - p[t][i-1][j-1];

}

while(q--){

d=0;

cin>>t>>x1>>y1>>x2>>y2;

for(int i=1; i<=c; i++){

d+= (p[i][x2][y2] - p[i][x1-1][y2] - p[i][x2][y1-1] + p[i][x1-1][y1-1] ) \* ((i+t%(c+1))%(c+1));

}

cout<<d<<"\n";

}

}

1. **Đánh giá độ phức tạp**

+ Vòng lặp đầu tiên: O(n)

+ Vòng lặp thứ 2: O(104 \* c)

+ Vòng lặp thứ 3: O(q \* c)

=> Độ phức tạp của thuật toán là: O(n)