

原创 The Preliminary Contest for ICPC Asia Nanjing 2019 A. The beautiful values of the p (扫描线)

2019-09-02 08:44:57 _-Y-_-Y- 阅读数 42 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_44410512/article/details/100191087

The beautiful values of the palace

The beautiful values of the palace

本题一个思维点 通过 X, Y 的坐标求出相应数字，可以通过分类讨论，或者推公式找出结论

地图大小为 $n \times n$ n 一定为奇数

$$x = x - n/2 - 1;$$

$$y = x - n/2 - 1$$

$$t = \max(\text{abs}(x), \text{abs}(y)); \text{ //确定该点在第几圈螺旋}$$

$$\text{if}(x \geq y) \text{ans} = N * N - 4 * t * t - 2 * t - x - y; \text{ //在向右与向上的路线上}$$

$$\text{elseans} = N * N - 4 * t * t + 2 * t + x + y; \text{ //在向左与向下的路线上}$$

对于这个题而言

就转化为求任意子矩阵之和

离散化 宫殿和询问的y坐标

对于每个询问要拆成4次询问

$\text{map}[i][j]$ 代表 1, 1 这点为左下节点 i, j 为右上节点所求得矩阵和

对于查询 $x1, y1, x2, y2$

$$\text{ans} = \text{map}[x2][y2] - \text{map}[x2][y1 - 1] - \text{map}[x1 - 1][y2] + \text{map}[x1 - 1][y1 - 1];$$

因此一次询问就可以看出求四次二维前缀和 标记每次询问是加还是减；

将宫殿和询问按 x 坐标排序

维护一个带修树状数组

每加入一个宫殿添加其值

每加入个询问就求其前缀和

https://blog.csdn.net/weixin_44410512

Code

```
1  /*
2  The beautiful values of the palace
3  The Preliminary Contest for ICPC Asia Nanjing 2019 A.
4  https://nanti.jisuanke.com/t/41298
5  题意： 有一个n*n (n为奇数)的矩阵，沿着方阵中心螺旋下降，有m个地方是要建宫殿的，
6  宫殿的价值是土地价值的数字总和 例如：123213 (1+2+3+2+1+3=12)
7  p次询问 问一个区间的所以宫殿价值总和
8  解法：因为n比较大，所以只能用数学方法计算出要建宫殿的地方的价值
9  然后因为p比较大，所以要用到扫描线来处理
10 */
11 #include <bits/stdc++.h>
12 using namespace std;
13 #define maxn 1000100
14 #define ll long long
```

猿衣酷

专属于程序员的卫衣

关闭

```

15 int n,m,p,cnt;
16 int tree[maxn];
17 int sum[maxn],res[maxn];
18 struct node{
19     int f,L,R,h,id;
20     int val;
21 }edge[maxn];
22 int get(int x, int y){
23     long long centre = (long long)n*n;
24     long long cen = n/2+1;
25     long long val;
26     if (x==cen && y==cen){
27         int ans=0;
28         val=centre;
29         while(val){
30             ans+=val%10;
31             val=val/10;
32         }
33         return ans;
34     }
35     long long Max = max(abs(x-cen), abs(y-cen));
36     val = centre - Max * (4*Max +1);
37     if((y == (cen-Max) && x >= cen) || x == cen + Max) val -= (x-cen) + (y-cen+Max);
38     else {
39         if(y != (cen-Max)) val += (x-cen + Max) + (y-cen + Max) + Max;
40         else val += cen - x;
41     }
42     int ans=0;
43     while(val){
44         ans+=val%10;
45         val=val/10;
46     }
47     return ans;
48 }
49 bool cmp(node a,node b){
50     if(a.h==b.h){
51         if((a.f==1&&b.f==0)||(b.f==1&&a.f==0)) return a.f>b.f;
52         else return a.f<b.f;
53     }
54     return a.h<b.h;
55 }
56 void update(int pow,int val){
57     while(pow<maxn){
58         tree[pow]+=val;
59         pow+=(-pow)&pow;
60     }
61 }
62 int query(int pow){
63     int ans=0;
64     while(pow>0){
65         ans+=tree[pow];
66         pow-=(-pow)&pow;
67     }
68     return ans;
69 }
70 int main(){
71     int T;
72     int x1,x2,y1,y2,x,y;
73     cin>>T;
74     while(T--){
75         scanf("%d %d %d", &n, &m, &p);
76         cnt=0;
77         memset(tree,0,sizeof tree);
78         while(m--){
79             scanf("%d %d", &x, &y);
80             edge[cnt++]=(node){0,x,0,y,0,get(x,y)};
81         }
82         for(int i=1;i<=p;i++){
83             scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
84             edge[cnt++]=(node){1,x1,x2,y1,i,0};
85             edge[cnt++]=(node){2,x1,x2,y2,i,0};

```

猿衣酷

专属于程序员的卫衣

关闭

```
86     }
87     sort(edge,edge+cnt,cmp);
88     for(int i=0;i<cnt;i++){
89         if(edge[i].f==0) update(edge[i].L,edge[i].val);
90         else if(edge[i].f==1) sum[edge[i].id]=query(edge[i].R)-query(edge[i].L-1);
91         else res[edge[i].id]=query(edge[i].R)-query(edge[i].L-1)-sum[edge[i].id];
92     }
93     for(int i=1;i<=p;i++){
94         printf("%d\n", res[i]);
95     }
96 }
97 return 0;
98 }
99 }
```

有 0 个人打赏

文章最后发布于: 201

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客

猿衣酷

专属于程序员的卫衣

关闭