

D —— 解题报告

题意:给出一个 $R \times C$ 的矩阵,矩阵中每个点有值,再给出 Q 次查询,询问左上角为 $(r1,c1)$,右下角为 $(r2,c2)$ 的小矩阵中的最小值.

思路:一眼就能看出是 `rmq`,使用 `ST` 算法解决.由于 $R \times C \leq 1000000$,所以二维 `rmq` 可能开不下,但我们可以想到 $\min(R,C) \leq \sqrt{R \times C} \leq 1000$,于是我们可以针对 R 和 C 中比较小的一个做一维 `rmq`,即将矩阵分成一个个的行或列,每次查询子矩阵的时候,求该子矩阵中所有行或列的最小值中的最小值即可.这样就降低了空间复杂度.

```
1 #include <cstdio>
2 #include <cstring>
3 #include <cmath>
4 #include <algorithm>
5 #include <vector>
6
7 using namespace std;
8
9 const int N = 1000010;
10 const int inf = 0x7fffffff;
11
12 int Log[N],g[N];
13 int R,C,Q;
14 vector<vector<vector<int>>>> rmq;
15
16 inline void setV(int r,int c,int v)
17 {
18     g[r*C+c]=v;
19 }
20 inline int getV(int r,int c)
21 {
22     return g[r*C+c];
23 }
24 void init(void)
25 {
26     rmq.clear();
27     for(int i=0;i<R;i++)
28     {
29         rmq.push_back(vector<vector<int>>>());
30         for(int k=0;(1<=k)-1<C;k++)
31         {
32             rmq[i].push_back(vector<int>());
33             for(int j=0;(1<=k)-1<C;j++)
34                 rmq[i][k].push_back(k==0?getV(i,j):min(rmq[i][k-1][j],rmq[i][k-1][j+(1<=k-1)]));
35         }
```

```

36     }
37 }
38 inline int query(int r,int c1,int c2)
39 {
40     int k=Log[c2-c1+1];
41     return min(rmq[r][k][c1],rmq[r][k][c2-(1<<k)+1]);
42 }
43 int main(void)
44 {
45     for(int i=1;i<N;i++)    Log[i]=(int)log2((double)i);
46     while(scanf("%d%d%d",&R,&C,&Q)==3 && (R||C||Q))
47     {
48         bool rotate=(R>C);
49         int r=R,c=C;
50         if(rotate)    swap(R,C);
51         for(int i=0;i<r;i++)    for(int j=0;j<c;j++)
52         {
53             int v;
54             scanf("%d",&v);
55             if(rotate)    setV(j,i,v);
56             else    setV(i,j,v);
57         }
58         init();
59         for(int i=0;i<Q;i++)
60         {
61             int r1,c1,r2,c2;
62             scanf("%d%d%d%d",&r1,&c1,&r2,&c2);
63             if(rotate)    swap(r1,c1),swap(r2,c2);
64             int v=inf;
65             for(int j=r1;j<=r2;j++)
66                 v=min(v,query(j,c1,c2));
67             printf("%d\n",v);
68         }
69     }
70     return 0;
71 }

```