

## 原创 ACM-ICPC 2018 徐州赛区网络预赛 J. Maze Designer (最大生成树+LCA)

2019-08-30 09:59:35 \_Y-\_Y\_ 阅读数 18 更多

编辑

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/100151333](https://blog.csdn.net/weixin_44410512/article/details/100151333)

### Maze Designer

最大生成树 + LCA

比赛的时候完全没想到。。。补题系列。。。

### AC code

```
1  /*
2  Maze Designer
3  ACM-ICPC 2018 徐州赛区网络预赛 J.
4  https://nanti.jisuanke.com/t/A2009
5  题意：有一个n*m的空地，可以在每个空地之间建墙，多个询问，问从某一点到另外一点的有且只有一条路并且建墙的总代价最小时的路径长度
6  思路：假设开始所有的墙都已经建好了，现在开始拆墙，保证拆的墙一定是最大的，然后算两点之间的距离
7  解法：拆墙就是找最大生成树，算两点之间的距离就哟个LCA 因为有多组询问
8  */
9  #include <bits/stdc++.h>
10 using namespace std;
11 #define maxn 700000
12 struct node{
13     int u,v,w;
14     int next;
15 }edge[maxn],edgek[maxn];
16 int head[maxn],fa[maxn],deep[maxn];
17 int anc[maxn][25];
18 int cnt,cntk,n,m,k,q;
19 char c;
20 void init(){
21     memset(head,-1,sizeof head);
22     cnt=cntk=0;
23 }
24 void add(int u,int v,int w){
25     edge[cnt].u=u;
26     edge[cnt].v=v;
27     edge[cnt].w=w;
28     edge[cnt].next=head[u];
29     head[u]=cnt++;
30 }
31 int root(int x){
32     return fa[x]==x?x:fa[x]=root(fa[x]);
33 }
34 void unite(int x,int y){
35     x=root(x);
36     y=root(y);
37     if(x!=y)
38         fa[y]=x;
39 }
40 int alike(int x,int y){
41     return root(x)==root(y);
42 }
43 bool cmp(node a,node b){
44     return a.w>b.w;
45 }
46 void kruskal(){
47     for(int i=0;i<=(n*m);i++) fa[i]=i;
48     sort(edgek,edgek+cntk,cmp);
49     int num=0;
50     for(int i=0;i<cntk;i++){
51         if(!alike(edgek[i].u,edgek[i].v)){
```

猿衣酷

专属于程序员的卫衣

关闭

```

52         unite(edgek[i].u, edgek[i].v);
53         num++;
54         add(edgek[i].u, edgek[i].v, edgek[i].w);
55         add(edgek[i].v, edgek[i].u, edgek[i].w);
56     }
57 }
58 }
59 void dfs(int u, int fa) {
60     anc[u][0] = fa;
61     for (int i = 1; i <= 22; i++) {
62         anc[u][i] = anc[anc[u][i-1]][i-1];
63     }
64     for (int i = head[u]; ~i; i = edge[i].next) {
65         int v = edge[i].v;
66         if (v != fa) {
67             deep[v] = deep[u] + 1;
68             dfs(v, u);
69         }
70     }
71 }
72 void solve() {
73     deep[0] = 1;
74     dfs(0, 0);
75 }
76 int lca(int x, int y) {
77     if (deep[x] < deep[y]) swap(x, y);
78     for (int i = 22; i >= 0; i--) {
79         if (deep[y] <= deep[anc[x][i]]) {
80             x = anc[x][i];
81         }
82     }
83     if (x == y) return x;
84     for (int i = 22; i >= 0; i--) {
85         if (anc[x][i] != anc[y][i]) {
86             x = anc[x][i];
87             y = anc[y][i];
88         }
89     }
90     return anc[x][0];
91 }
92 int main() {
93     cin >> n >> m;
94     init();
95     for (int i = 0; i < n; i++) {
96         for (int j = 0; j < m; j++) {
97             cin >> c >> k;
98             if (c == 'D') {
99                 edgek[cntk].u = i * m + j;
100                 edgek[cntk].v = i * m + m + j;
101                 edgek[cntk++].w = k;
102             }
103             cin >> c >> k;
104             if (c == 'R') {
105                 edgek[cntk].u = i * m + j;
106                 edgek[cntk].v = i * m + 1 + j;
107                 edgek[cntk++].w = k;
108             }
109         }
110     }
111     kruskal();
112     solve();
113     cin >> q;
114     int u1, v1, u2, v2, u, v;
115     while (q--) {
116         cin >> u1 >> v1 >> u2 >> v2;
117         u = (u1 - 1) * m + v1 - 1;
118         v = (u2 - 1) * m + v2 - 1;
119         cout << deep[u] + deep[v] - 2 * deep[lca(u, v)] << endl;
120     }
121     return 0;
122 }

```

猿衣酷

专属于程序员的卫衣

关闭

有 0 个人打赏

文章最后发布于: 2019-08-30 09:59:35

©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