

## F —— 解题报告

题意:给出  $n$  个点  $m$  条边的无向图,每个点有点权,在图上定义三种操作:

- 1.查询所有与  $X$  点相连的点中,点权不小于  $K$  的最小点权
- 2.把  $X$  的点权改为  $K$
- 3.从图中删除  $A$  到  $B$  的边

求所有操作 1 的结果的平均值 (即所有操作 1 所得的点权和/操作 1 执行的次数)

思路:首先建图,然后删边,再逆向处理所有操作,即可将删边变为加边,每次将小的集合合并到大的中,复杂度  $N \cdot \log N \cdot \log N$ ,最后输出.集合用 `set` 实现,并路径压缩.

```
1 #include <cstdio>
2 #include <cstring>
3 #include <set>
4 #include <algorithm>
5
6 using namespace std;
7
8 const int N=20010;
9 const int E=60010;
10 const int Q=300010;
11
12 int n,m,q,cas=1;
13 int val[N],que[Q][3];
14 multiset<pair<int,int>>e;
15 multiset<pair<int,int>>::iterator it;
16 multiset<int> v[N];
17 multiset<int>::iterator itt;
18 int f[N];
19
20 inline void in it(void)
21 {
22     e.clear();
23     for(int i=1;i<=n;i++)
24     {
25         scanf("%d",val+i);
26         v[i].clear();
27         f[i]=i;
28     }
29 }
30 int find(int x)
31 {
32     if(f[x]!=x) f[x]=find(f[x]);
33     return f[x];
34 }
35 void Union(int x,int y)
```

```

36 {
37     x=find(x),y=find(y);
38     if(x==y) return;
39     if(v[x].size()>v[y].size()) swap(x,y);
40     v[y].insert(v[x].begin(),v[x].end());
41     v[x].clear();
42     f[x]=y;
43 }
44 void update(int x,int y)
45 {
46     int tx=x;
47     x=find(x);
48     v[x].erase(v[x].find(val[tx]));
49     v[x].insert(y);
50     val[tx]=y;
51 }
52 int main(void)
53 {
54     while(scanf("%d%d",&n,&m,&q)==3)
55     {
56         init();
57         for(int i=0;i<m;i++)
58         {
59             int u,v;
60             scanf("%d%d",&u,&v);
61             if(u>v) swap(u,v);
62             e.insert(make_pair(u,v));
63         }
64         for(int i=0;i<q;i++)
65         {
66             char opt[9];
67             scanf("%s",opt);
68             scanf("%d%d",&que[i][1],&que[i][2]);
69             que[i][0]=opt[0];
70             if(opt[0]=='E')
71             {
72                 if(que[i][1]>que[i][2]) swap(que[i][1],que[i][2]);
73                 e.erase(e.find(make_pair(que[i][1],que[i][2])));
74             }
75             else if(opt[0]=='U') swap(val[que[i][1]],que[i][2]);
76         }
77         for(int i=1;i<=n;i++) v[i].insert(val[i]);
78         for(it=e.begin();it!=e.end();it++)
79             Union(it->first,it->second);

```

```
80     double cnt=0,sum=0;
81     for(int i=q-1;i>=0;i--)
82     {
83         if(que[i][0]=='E')    Union(que[i][1],que[i][2]);
84         else if(que[i][0]=='U')    update(que[i][1],que[i][2]);
85         else
86         {
87             cnt+=1;
88             int u=find(que[i][1]);
89             itt=v[u].lower_bound(que[i][2]);
90             if(itt!=v[u].end())    sum+=*itt;
91         }
92     }
93     printf("Case %d: %.3lf\n",cas++,sum/cnt);
94 }
95 return 0;
96 }
```