

# 题意

此题在bzoj为权限题，cf 920E题目相同

给出无向图的点数 $n$ ，序对数 $m$  ( $1 \leq n \leq 200000, 0 \leq m \leq \min(\frac{n(n-1)}{2}, 200000)$ )

给出 $m$ 个序对  $(x,y)$  表示点 $x$ 和点 $y$ 间，没有边相邻

求该无向图有几个连通块，并按升序输出每个连通块的大小

# 分析

由题意，若 $x,y$ 在同一连通块，则序对  $(x,y)$ 不存在，即已知原图求补图的连通块

普通求连通块常用BFS方法：每次枚举一个未访问过的点，从它开始宽搜出它所在连通块的所有点

这里可以，枚举它原图的边，标记相邻点。再枚举所有的点，将未标记的点未访问过的点加入当前连通块，并加入队列

考虑优化：

如果一个点已经加入过连通块，就不必再枚举了，用链表维护，如果一个点入队，则删去这个点，那么枚举所有点那一步就可以遍历这个链表

(我的代码效率有点低)

# 代码

```
1 #include<bits/stdc++.h>
2 #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
3 #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
4 #define Memset(a,b); memset((a),(b),sizeof((a)));
5 #define Cin(a); scanf("%d",&(a));
6 #define Cinc(a); scanf(" %c",&(a));
7 #define Cins(a); scanf("%s",(a));
8 #define Cout(a,b); printf("%d",(a));printf(b);
9 #define Coutc(a,b); printf("%c",(a));printf(b);
10 #define Couts(a,b); printf("%s",(a));printf(b);
11 using namespace std;
12 typedef long long LL;
13 typedef unsigned long long ULL;
14 typedef long double LDB;
15 inline int readint() {int x;cin>>x;return x;}
16 const int N = 2e5+5;
17 struct cre{
18     int nex,pre;
```

```

19 }table[N];
20 int n,m;
21 bool belong[N];
22 int ltk[N],tot;
23 vector<int> v[N];
24 int que[N],fro,tai;
25 bool book[N];
26 int main()
27 {
28     ios::sync_with_stdio(false);
29     cin>>n>>m;
30     int x,y;
31     For(i,1,m)
32     {
33         cin>>x>>y;
34         v[x].push_back(y);
35         v[y].push_back(x);
36     }
37     For(i,1,n)
38     {
39         table[i].pre = i-1;
40         table[i].nex = i+1;
41     }
42     table[0].nex = 1;
43     table[n+1].pre = n;
44     int use = 0;
45     while(use<n)
46     {
47         ++tot;
48         fro = 0;
49         tai = 0;
50         int t = table[0].nex;
51         que[tai++] = t;
52         belong[t] = true;
53         table[table[t].pre].nex = table[t].nex;
54         table[table[t].nex].pre = table[t].pre;
55         ltk[tot]++;
56         ++use;
57         while(fro != tai)
58         {
59             int now = que[fro];
60             fro++;
61             Memset(book,false);
62             book[now] = true;
63             for(int i=0;i<v[now].size();i++)
64                 book[v[now][i]] = true;

```

```
65         int t = table[0].nex;
66         while(t<=n)
67         {
68             if(book[t] == false && belong[t] == false)
69             {
70                 ++use;
71                 belong[t] = true;
72                 ltk[tot]++;
73                 que[tai++] = t;
74                 table[table[t].pre].nex = table[t].nex;
75                 table[table[t].nex].pre = table[t].pre;
76             }
77             t = table[t].nex;
78         }
79     }
80 }
81 sort(ltk+1, ltk+tot+1);
82 cout<<tot<<endl;
83 For(i, 1, tot)
84     cout<<ltk[i]<<" ";
85 cout<<endl;
86 }
```