

# 牛客多校第八场

想起高中时看的一本大黑砖：

《我怎样爆零[HowDoIGetZero]

所以预计本周周报将陷入无题可写的困境。

比赛时以为是二分图的匹配用匈牙利算法做了，搞了半天搞出来了但是复杂度太高了过不去。

参考题解的答案是：总点数-#[]k个点[] k-1条边的连通分量)

```
#include<stdio.h>
#include<string.h>

#include<vector>
#include<algorithm>

using namespace std;

bool vis[200020];
int F[200020];

void init()
{
    memset(F,-1,sizeof(F));
    memset(vis,0,sizeof(vis));
}

int find(int x)
{
    if(F[x]==-1)
    {
        return x;
    }
    return F[x]=find(F[x]); //改了根
}

void bing(int x,int y)
{
    int t1=find(x); //找到x的根
    int t2=find(y); //找到y的根
    if(t1==t2)
    {
        vis[t1]=1; //同一个根，该根变1
        return;
    }
}
```

```
    }
    F[t1]=t2;//如果x和y的根不同,修改t1的根t2
    if(vis[t1])
    {
        vis[t2]=1;
    }
}

int main()
{
    int t;
    scanf("%d",&t);
    int k=0;
    while(t--)
    {
        k++;
        init();
        int n;
        scanf("%d",&n);
        vector<int> a(n),b(n);
        vector<int> val;
        int i;
        for(i=0;i<n;i++)
        {
            scanf("%d%d",&a[i],&b[i]);
            val.push_back(a[i]);
            val.push_back(b[i]);
        }
        sort(val.begin(),val.end());
        int tot=unique(val.begin(),val.end())-val.begin();
        val.resize(tot);
        for(i=0;i<n;i++)
        {
            a[i]=lower_bound(val.begin(),val.end(),a[i])-val.begin();
            b[i]=lower_bound(val.begin(),val.end(),b[i])-val.begin();
            bing(a[i],b[i]);
        }
        int ans=tot;
        for(i=0;i<tot;i++)//遍历c中不同的点
        {
            if(F[i]==-1&&!vis[i])
            {
                ans--;
            }
        }
        printf("Case # %d: %d\n",k,ans);
    }
}
```

From:  
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:  
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E7%89%9B%E5%AE%A2%E5%A4%9A%E6%A0%A1%E7%AC%AC%E5%85%AB%E5%9C%BA&rev=1596875721> 

Last update: **2020/08/08 16:35**