思路：

想一想如果把直接或者间接存在关系的人分成若干个集合，对于每个集合来说，是不是只要花费一个1W买其中一个人，剩下的人全都可以优惠，最多能优惠多少呢？是不是就是求这个集合的最大生成树？，so 问题就解决了。ans=(n+m)\*10000-最大生成树（森林）

#include<cstdio>

#include<algorithm>

using namespace std;

typedef long long ll;

struct node

{

int boy,girl,v;

}a[60010];

bool cmp(node a,node b)

{

return a.v>b.v;

}

int fa[40010];

int Find(int x)

{

if(fa[x]==x)return x;

return fa[x]=Find(fa[x]);//是否需要带路径压缩还要考虑一下,有时带了会超时

}

int main()

{

//freopen("input.txt","r",stdin);

int T,n,m,r;

scanf("%d",&T);

while(T--)

{

scanf("%d%d%d",&n,&m,&r);

for(int i=0;i<r;i++)

{

scanf("%d%d%d",&a[i].girl,&a[i].boy,&a[i].v);

a[i].boy+=n;

}

sort(a,a+r,cmp);

for(int i=0;i<n+m;i++)

fa[i]=i;

ll ans=(n+m)\*10000;

for(int i=0;i<r;i++)

{

int u=Find(a[i].boy);

int v=Find(a[i].girl);

if(u!=v)

{

ans-=a[i].v;

fa[v]=u;

}

}

printf("%lld\n",ans);

}

return 0;

}

法2：

#include<cstdio>

#include<algorithm>

#include<cstring>

#include<iostream>

using namespace std;

typedef long long ll;

/\*

\* Kruskal算法求MST

\*/

const int MAXN=21000;//最大点数

const int MAXM=51000;//最大边数

int F[MAXN];//并查集使用

struct Edge

{

int u,v,w;

} edge[MAXM]; //存储边的信息，包括起点/终点/权值

int tol;//边数，加边前赋值为0

bool cmp(Edge a,Edge b)

{

//排序函数，讲边按照权值从小到大排序

return a.w<b.w;

}

int find(int x)

{

if(F[x]==-1)return x;

else return F[x]=find(F[x]);

}

int Kruskal(int n)//传入点数，返回最小生成树的权值，如果不连通返回-1

{

memset(F,-1,sizeof(F));

sort(edge,edge+tol,cmp);

int ans=0;

for(int i=0; i<tol; i++)

{

int u=edge[i].u;

int v=edge[i].v;

int w=edge[i].w;

int t1=find(u);

int t2=find(v);

if(t1!=t2)

{

ans+=w;

F[t1]=t2;

}

}

return ans;

}

int main()

{

//freopen("input.txt","r",stdin);

int T,n,m,r;

scanf("%d",&T);

while(T--)

{

scanf("%d%d%d",&n,&m,&r);

for(int i=0; i<r; i++)

{

scanf("%d%d%d",&edge[i].u,&edge[i].v,&edge[i].w);

edge[i].v+=n;

edge[i].w=-edge[i].w;

}

ll ans=(n+m)\*10000;

tol=r;

printf("%lld\n",ans+Kruskal(n+m));

}

return 0;

}