

我们都是小青呱，WA WA WA WA WA!

有很多种方法能做出这道题，这里展示其中一种。

因为桥梁的种类最多只有 10 种，可以考虑对种类进行枚举。枚举每个种类的桥梁的使用与不使用的情况，并使用并查集（或者搜索）的方式判断每种情况下 1 号岛屿和 n 号岛屿是否连通，复杂度为 $O(2^W n)$ ，其中 W 为桥梁种类数。

注意本题空间卡常，如果使用二维数组可能需要开 *bool*。

```
#include<bits/stdc++.h>
#define int long long
#define Nmax 10010
#define IOS ios::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr);
using namespace std;
int n,m;
struct EDGE
{
    int u,v,w;
}edge[Nmax];
struct DSU
{
    int fa[Nmax];
    void initial()
    {
        for(int i=1;i<=n;i++)
            fa[i]=i;
    }
    int find(int x)
    {
        if(fa[x]==x)
            return x;
        return fa[x]=find(fa[x]);
    }
    void merge(int x,int y)
    {
        x=find(x);
        y=find(y);
        if(x>y)
            fa[x]=y;
        else
            fa[y]=x;
    }
}dsu;
signed main()
{
    IOS;
    cin>>n>>m;
    for(int i=1;i<=m;i++)
        cin>>edge[i].u>>edge[i].v>>edge[i].w;
    for(int i=1;i<(1ll<<10);i++)
    {
        dsu.initial();
        for(int j=1;j<=m;j++)
            if((1ll<<(edge[j].w-1)&i)
                dsu.merge(edge[j].u,edge[j].v);
        if(dsu.find(1)==dsu.find(n))
        {
            cout<<i;
            exit(0);
        }
    }
}
```

```
}  
cout<<-1;  
}
```