



最小生成树 (MST) —— Kruskal

贪心策略：每次选择权重最小的边加入生成树，同时避免形成环路

可用于解决最小网络，最低价联通之类的问题

例题

P3366 【模板】最小生成树

```
#include<bits/stdc++.h>
#define int long long
const int maxn = 2e5+5;
int n,m,ans,cnt;
struct Edge{
    int u,v,w;
    friend bool operator <(Edge a,Edge b){
        return a.w<b.w;
    }
} e[maxn];

namespace dsu{
    int fa[maxn];
    inline void init(int x){for(int i=1;i<=x;i++)fa[i] = i;}
    inline int find(int a){return fa[a]==a?a:fa[a] = find(fa[a]);}
    inline void merge(int a,int b){fa[find(a)] = find(b);}
    inline bool check(int a,int b){return find(a)==find(b);}
}

signed main(){
    scanf("%d%d",&n,&m);
    for(int i=1;i<=m;i++)scanf("%d%d%d",&e[i].u,&e[i].v,&e[i].w);
    std::sort(e+1,e+1+m);
    dsu::init(n);
    for(int i=1;i<=m;i++){
        if(!dsu::check(e[i].u,e[i].v)){//避免环路
            dsu::merge(e[i].u,e[i].v);
            ans+=e[i].w;
            cnt++;
        }
    }
    if(cnt==n-1)printf("%d\n",ans);
    else puts("orz");//无法生成最小生成树 -> 原图不是DAG
    return 0;
}
```