

最大流(Dinic)

```
#include <bits/stdc++.h>
using namespace std;
#define rep(i,a,b) for(int i=a;i<=b;i++)
#define per(i,a,b) for(int i=a;i>=b;i--)
typedef long long ll;
const int MAXN=300100;
const int MAXM=300100;
const ll oo=0x7fffffffffff;
inline int read() {
    int x=0,f=1;
    char c=getchar();
    while(c<'0' || c>'9'){if(c=='-') f=-1;c=getchar();}
    while(c>='0' && c<='9') x=x*10+c-'0',c=getchar();
    return x*f;
}
struct node{
    int to,nxt;
    ll w;
}ed[MAXM*2];
int n,m,s,t,d,u,v,tot=1,head[MAXN],cur[MAXN],in[MAXN];
ll w,dis[MAXN],leaf[MAXN];
ll mxflow;
queue<int>q;
void add(int u,int v,ll w){
    ed[++tot].to=v,ed[tot].w=w,ed[tot].nxt=head[u],head[u]=tot;
    ed[++tot].to=u,ed[tot].w=0,ed[tot].nxt=head[v],head[v]=tot;
}
bool bfs(){
    memset(dis,0,sizeof(dis));
    q.push(s),dis[s]=1;
    while(!q.empty()){
        int u=q.front();q.pop();
        for(int i=head[u];i;i=ed[i].nxt){
            int v=ed[i].to;
            if(dis[v] || !ed[i].w) continue;
            q.push(v),dis[v]=dis[u]+1;
        }
    }
    return dis[t]==0?false:true;
}
ll dfs(int u,ll flow){
    if(u==t){
        mxflow+=1ll*flow;
        return flow;
    }
    long long used=0;
    for(int &i=cur[u];i;i=ed[i].nxt){
        int v=ed[i].to;
        if(dis[v]==dis[u]+1){
            ll lft=dfs(v,min(flow-used,ed[i].w));
            if(lft) ed[i].w-=lft,ed[i^1].w+=lft,used+=lft;
            if(used==flow) return flow;
        }
    }
    return 0;
}
```

```

    }
}
return used;
}

int main(){
    n=read();m=read();s=read();t=read();
    rep(i,1,m){
        u=read();v=read();w=read();
        add(u,v,w);
    }
    while (bfs()){
        memcpy(cur,head,sizeof(head));
        dfs(s,oo);
    }
    cout << mxflow;
    return 0;
}

```