

```

1  /*
2     Author:Frozenscode
3  */
4  #include <bits/stdc++.h>
5  using namespace std;
6  #define pb push_back
7  #define mp make_pair
8  #define fi first
9  #define se second
10 typedef long long ll;
11 typedef pair<ll,ll> pii;
12 const ll maxn = 10010;
13 const ll INF = 2147483647;
14 ll n, m, s, t, tot = 1, head[maxn], nxt[maxn], now[maxn], x, y, v;
15 ll dis[maxn], res;
16 struct e{
17     ll to, val;
18 }edge[maxn];
19 void add_edge(ll x, ll y, ll val){
20     nxt[++tot] = head[x];
21     edge[tot].to = y;
22     edge[tot].val = val;
23     head[x] = tot;
24 }
25 bool bfs(){
26     for(int i = 1; i <= n; i++) dis[i] = INF;
27     queue <ll> q;
28     dis[s] = 0;
29     now[s] = head[s];
30     q.push(s);
31     while(!q.empty()){
32         ll tem = q.front();
33         //cout << tem << endl;
34         q.pop();
35         for(int i = head[tem]; i; i = nxt[i]){
36             if(dis[edge[i].to] == INF && edge[i].val){
37                 //cout << edge[i].to << endl;
38                 dis[edge[i].to] = dis[tem] + 1;
39                 now[edge[i].to] = head[edge[i].to];
40                 q.push(edge[i].to);
41                 if(edge[i].to == t) return true;
42             }
43         }
44     }
45     return false;
46 }
47 ll dfs(ll x, ll sum){
48     if(x == t) return sum;
49     ll ans = 0;
50     for(int i = now[x]; i && sum; i = nxt[i]){

```

```

51     now[x] = i;
52     if(dis[edge[i].to] == dis[x] + 1 && edge[i].val){
53         ll k = dfs(edge[i].to, min(sum, edge[i].val));
54         if(k == 0) dis[edge[i].to] = INF;
55         edge[i].val -= k;
56         edge[i ^ 1].val += k;
57         ans += k;
58         sum -= k;
59     }
60 }
61 return ans;
62 }
63 int main()
64 {
65     ios::sync_with_stdio(false);
66     cin >> n >> m >> s >> t;
67     for(int i = 1; i <= m; i++){
68         cin >> x >> y >> v;
69         add_edge(x, y, v);
70         add_edge(y, x, 0);
71     }
72     while(bfs()){
73         res += dfs(s, INF);
74     }
75     cout << res << endl;
76     return 0;
77 }

```