///code by sonhotrung

///2 dòng này sẽ mất:)))

#include <bits/stdc++.h>

#define pb push\_back

#define pii pair<int,int>

#define fi first

#define se second

const int maxn=1e3+5;

const long long oo=1e18;

using namespace std;

int n,m,s,t;

long long delta;

vector <int> E[maxn];

int tr[maxn];

long long C[maxn][maxn],F[maxn][maxn];

void BFS(int s)

{

queue <int> Q;

Q.push(s);

while (!Q.empty())

{

int u=Q.front();

Q.pop();

for (int v:E[u])

if (tr[v]==0&&C[u][v]>F[u][v])

{

tr[v]=u;

Q.push(v);

}

}

}

void DFS(int u)

{

for (int v:E[u])

if (tr[v]==0&&C[u][v]>F[u][v])

{

tr[v]=u;

DFS(v);

}

}

bool Find\_path()

{

memset(tr,0,sizeof(tr));

tr[s]=-1;

BFS(s);

return tr[t];

}

void Inc\_Flow()

{

delta=oo;

int v=t;

while (v!=s)

{

int u=tr[v];

delta=min(delta,C[u][v]-F[u][v]);

v=u;

}

v=t;

while (v!=s)

{

int u=tr[v];

F[u][v]+=delta;

F[v][u]-=delta;

v=u;

}

}

int main()

{

//freopen("","r",stdin);

//freopen("","w",stdout);

ios\_base::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

cin >> n >> m >> s >> t;

while (m--)

{

int u,v,c;

cin >> u >> v >> c;

E[u].pb(v);

E[v].pb(u);

C[u][v]=c;

}

long long maxflow=0;

while (Find\_path())

{

Inc\_Flow();

maxflow+=delta;

}

cout << maxflow;

return 0;

}