#include<bits/stdc++.h>

using namespace std;

vector<vector<pair<int,int> > > graph;

void addEdge(int x, int y, int w){

graph[x].push\_back(make\_pair(y,w));

graph[y].push\_back(make\_pair(x,w));

}

void display(){

for(int i=0;i<graph.size();i++){

cout<<i<<"->";

for(auto it: graph[i]){

cout<<it.first<<" "<<it.second<<", ";

}

cout<<endl;

}

}

//2@2

// 1@21

// 3@23

// 0@210

// 4@234

// 5@2345

// 6@2346

typedef pair<int,int> pi;

int infected(int src, vector<bool> visited, int t){

queue<pi> q;

q.push(make\_pair(src,1));

visited[src] = true;

int count = 1;

while(!q.empty()){

pair<int,int> temp = q.front();

q.pop();

if(t==temp.second){

return count;

}

for(auto it: graph[temp.first]){

if(visited[it.first]==false){

q.push(make\_pair(it.first,temp.second+1));

visited[it.first] = true;

count++;

}

}

}

return count;

}

int main(){

int v,e;

cin>>v>>e;

graph.resize(v);

while(e--){

int x,y,w;

cin>>x>>y>>w;

addEdge(x,y,w);

}

int src,t;

cin>>src>>t;

vector<bool> visited(v);

cout<<infected(src,visited,t);

}