#include<bits/stdc++.h>

using namespace std;

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

int min\_ = INT\_MAX;

int max\_ = INT\_MIN;

int cmax\_ = INT\_MAX;

int fmin\_ = INT\_MIN;

string smin,smax,cstr,fstr,kstr;

typedef pair<int, string> pi;

priority\_queue<pi,vector<pi>,greater<pi> > pq;

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;

}

}

void allPath(int src, int dest, vector<bool> visited, string ans){

if(src==dest){

cout<<ans<<endl;

return;

}

visited[src] = true;

for(auto it: graph[src]){

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

string str = to\_string(it.first);

allPath(it.first,dest,visited,ans+str);

}

}

visited[src] = false;

}

void path\_wieght(int src, int dest, vector<bool> &visited, int wt, string ans, int criteria, int k){

if(src==dest){

if(min\_>wt){

min\_ = wt;

smin = ans;

}

if(max\_<wt){

max\_ = wt;

smax = ans;

}

if(wt>criteria && wt<cmax\_){

cmax\_ = wt;

cstr = ans;

}

if(wt<criteria && wt>fmin\_){

fmin\_ = wt;

fstr = ans;

}

if(pq.size()<k){

pq.push(make\_pair(wt,ans));

}

else{

pair<int,string> it = pq.top();

if(it.first<wt){

pq.pop();

pq.push(make\_pair(wt,ans));

}

}

return;

}

visited[src] = true;

for(auto it: graph[src]){

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

string str = to\_string(it.first);

path\_wieght(it.first,dest,visited,wt+it.second,ans+str,criteria,k);

}

}

visited[src] = false;

}

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,dest;

cin>>src>>dest;

// display();

vector<bool> visited(v);

string s = to\_string(src);

int criteria,k;

cin>>criteria>>k;

path\_wieght(src,dest,visited,0,s,criteria,k);

// Smallest Path = 01256@28

// Largest Path = 032546@66

// Just Larger Path than 30 = 012546@36

// Just Smaller Path than 30 = 01256@28

// 4th largest path = 03456@48

pair<int,string> it = pq.top();

cout<<"Smallest Path = "<<smin<<"@"<<min\_<<endl;

cout<<"Largest Path = "<<smax<<"@"<<max\_<<endl;

cout<<"Just Larger Path than "<<criteria<<" = "<<cstr<<"@"<<cmax\_<<endl;

cout<<"Just Smaller Path than "<<criteria<<" = "<<fstr<<"@"<<fmin\_<<endl;

cout<<k<<"th largest path = "<<it.second<<"@"<<it.first<<endl;

}