#include<bits/stdc++.h>

using namespace std;

// 0 1 bfs

bool arr[100000];

long dist[100000];

class edge

{

int wt;

int node;

public:

edge(int a,int b)

{

wt=a;node=b;

}

int retwt()

{

return(wt);

}

int retnode()

{

return(node);

}

};

vector<list<edge>>v;

void bfs(int s)

{

deque<int>q;

q.push\_front(s);

dist[s]=0;

arr[s]=true;

while(!q.empty())

{

int x=q.front();

q.pop\_front();

list<edge>:: iterator it;

for(it=v[x].begin();it!=v[x].end();it++)

{

if(arr[(\*it).retnode()]==false)

{

if((\*it).retwt()==0)

{

dist[(\*it).retnode()]=dist[x];

q.push\_front((\*it).retnode());

}

else

{

dist[(\*it).retnode()]=dist[x]+1;

q.push\_back((\*it).retnode());

}

arr[(\*it).retnode()]=true;

}

}

}

}

int main()

{

for(int i=0;i<100000;i++)

{

arr[i]=false;

}

for(int i=0;i<100000;i++)

{

dist[i]=-1;

}

int n, e;

cin >> n >> e;

v.assign(n+1, list<edge>());

for(int i=0;i<e;i++)

{

int a,b;

cin>>a>>b;

int weight;

cin>>weight;

v[a].push\_back(edge(weight,b));

}

bfs(3);

for(int i=1;i<=n;i++)

{

cout<<dist[i]<<" ";

}

return(0);

}