#include <bits/stdc++.h>

using namespace std;

#define huge 1e10;

int n,e;

class edge

{

public:

int wt;

int dest;

edge(int a,int b)

{

wt=a;

dest=b;

}

};

vector<list<edge>>v;

long dist[100006];

void dijkstra(int a)

{

dist[a]=0;

set<pair<long,int>>s;

s.insert(make\_pair(0,a));

while(!s.empty())

{

pair<long,int>temp=\*(s.begin());

s.erase(s.begin());

list<edge>::iterator it;

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

{

int w=(\*it).wt;

int val=(\*it).dest;

if(dist[val]>dist[temp.second]+w)

{

if (dist[val]!=1e10)

s.erase(s.find(make\_pair(dist[val], val)));

dist[val]=w+dist[temp.second];

s.insert(make\_pair(dist[val],val));

}

}

}

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

{

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

}

}

int main()

{

cin>>n>>e;

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

dist[i]=huge;

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

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

{

int s,d ,w;

cin>>s>>d>>w;

v[s].push\_back(edge(w,d));

v[d].push\_back(edge(w,s));

}

dijkstra(1);

return 0;

}