The problem statement

johnny works in a corporate company and has to travel between cities for his work, there are n cities and m flight connections between them. Your task is to determine the length of the shortest route from his city xx to every city.

Input

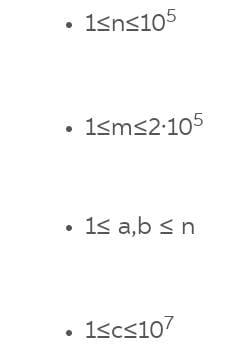
The first input line has two integers n(number of cities) and m (number of flight connections) The cities are numbered 1,2,…,n, and city 1 is his city xx.

then,m lines are describing the flight connections. Each line has three integers a(flight begins at city a), b(flight ends at city b), and c(its length is c), and each flight is a one-way flight.

Output

Print n integers: the shortest route lengths from xx to cities 1,2,…,n

Constraints



here is the code for the given problem:

//this solution uses graphs and dijktras algorithm

#include<bits/stdc++.h>

using namespace std;

#define int long long int

int n,m;

vector<vector<pair<int,int> > > v;

vector<int> dist;

const int MAXD=1e17;

//dijktras algorithm using priority queue

void solve(){

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

{

dist[i] = MAXD;

}

priority\_queue<pair<int,int>,

vector<pair<int,int>>,

greater<pair<int,int>> > pq;

dist[1] = 0;

pq.push({0,1});

while(!pq.empty())

{

int u = pq.top().second;

int d = pq.top().first;

pq.pop();

if(dist[u] < d) continue;

for(auto e:v[u]){

int a=e.first;

int b=e.second;

if(dist[a]<=d+b)continue;

else{

dist[a]=d+b;

pq.push({dist[a],a});

}

}

}

}

main(){

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

cin.tie(NULL);

cin>>n>>m;

v.resize(n+1);

dist.resize(n+1);

//building graph

for(int i=0;i<m;i++){

int a,b,c;

cin>>a>>b>>c;

v[a].push\_back({b,c});

}

solve();

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

{

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

}

}

Example

Input:

3 4

1 2 6

1 3 2

3 2 3

1 3 4

Output:

0 5 2