spfa

```
#include <bits/stdc++.h>
using namespace std;

const int N = 1e5 + 5, M = 5e5 * 2 + 10;
int n, m, s;
int tot, dist[N], h[N];
bool vis[N];
struct edge{
    int to, nxt, w;
}e[M];

void add(int u, int v, int w){
    e[++tot].to = v;
    e[tot].w = w;
    e[tot].nxt = h[u];
    h[u] = tot;
}

void spfa(){
    memset(dist, 0x3f, sizeof(dist));
    dist[s] = 0;
    queue<int> q;
    q.push(s);

    while(q.size()){
        int t = q.front();
        q.pop();
        vis[t] = false;
        for(int i = h[t]; i; i = e[i].nxt){
            int v = e[i].to;
            if(dist[t] + e[i].w < dist[v]){
                dist[v] = dist[t] + e[i].w;
                if(!vis[v]){
                    q.push(v);
                    vis[v] = true;
                }
            }
        }
    }
    return;
}

int main(){
    cin >> n >> m >> s;
    for(int i = 0; i < m; i++){
        int u, v, w;
        scanf("%d%d%d", &u, &v, &w);
        add(u, v, w);
    }
```

```
    spfa();

    for(int i = 1; i <= n; i++){
        printf("%d ", dist[i]);
    }
    return 0;
}
```