```cpp
#include <bits/stdc++.h>

#define read freopen("input.txt","r",stdin)
#define MAX 99999
#define MIN -1

using namespace std;

int min_value_arr[MAX];
int cost[MAX][3];
int dis[MAX];
int P[MAX];
int v, e, s;

void Initialize_Single_Source()
{
    for(int i=1; i<=v; i++)
    {
        dis[i] = MAX;
        P[i] = MIN;
    }
    dis[s] = 0;
}
int Weight(int u, int v)
{
    for(int i=1; i<=e; i++)
    {
        if(cost[i][0] == u && cost[i][1] == v)
        {
            return cost[i][2];
        }
    }
}
void Relax(int u , int v)
{
    if (dis[v] > dis[u] + Weight(u,v))
    {
        dis[v] = dis[u] + Weight(u,v);
        P[v] = u;
    }
}
int Extract_Min()
{
    int c = MAX , k;
    for(int i=1; i<=v; i++)
        if(dis[i] < c && !min_value_arr[i])
        {
            c = dis[i];
            k = i;
        }
    min_value_arr[k] = 1;
    return k;
}
void Dijkstra()
{
    Initialize_Single_Source();
    for(int i=1; i<=v; i++)
    {
        int m = Extract_Min();
        for(int j=1; j<=e; j++)
        {
            if (cost[j][0] == m)
                Relax(m,cost[j][1]);
        }
    }
}
```

```cpp

int main()
{
    cin >> v >> e >> s;
    for (int i=1; i<=e; i++)
    {
        cin >> cost[i][0] >> cost[i][1] >> cost[i][2];
    }
    Dijkstra();
    cout << endl << "Distance of the vertices : " << endl;
    for(int i=1; i<=v ; i++)
    {
        cout << dis[i] << "\t";
    }
    return 0;
}
```