

---

## **Assignment #2**

**1) Implement Bellman-Ford algorithm. Output should contain the shortest path, path cost, and the table containing the costs and parent of each node after each iteration.**

---

**Date of Performance: 18.06.2023**

**Date of Submission: 19.06.2023**

**Student ID: 20210104105**

**Name: MD. Manzar Nur Rahman**

**Group: C1**

```

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

struct Edge
{
    int u, v, w;
};

vector<Edge> E;
int cost[1000];
int parent[1000];
int n;

void printPath(int node)
{
    if (parent[node] == -1)
    {
        cout << node;
        return;
    }
    printPath(parent[node]);
    cout << " -> " << node;
}

void printTable()
{
    cout<<"DestinationNode\t\tParentNode\tcost\n"<<endl;
    for (int j = 1; j <= n; j++)
    {
        cout << j << "\t\t" << parent[j] << "\t\t" << cost[j] << "\n";
    }
}

void BellmanFord(int s)
{
    int cnt = 0;
    for (int i = 1; i <= n; i++)
    {
        cost[i] = 100000000;
        parent[i] = -1;
    }
    cost[s] = 0;
    cout << "\nAll Iterations: \n"<<endl;
    for (int i = 1; i < n; i++)
    {
        for (Edge e : E)

```

```

    {
        if (cost[e.v] > cost[e.u] + e.w)
        {
            cost[e.v] = cost[e.u] + e.w;
            parent[e.v] = e.u;
            cout << "on Iteration " << cnt << ":\n";
            printTable();
            cnt++;
        }
    }
}

int main()
{
    int node, edges;
    cout << "Enter the number of vetices: ";
    cin >> node;
    n = node;
    cout << "Enter the number of edges: ";
    cin >> edges;
    cout << "Enter the edges (source, destination, weight):\n";
    while (edges--)
    {
        Edge edge;
        cin >> edge.u >> edge.v >> edge.w;
        E.push_back(edge);
    }
    int source;
    cout << "Enter the source vertex: "<<endl;;
    cin >> source;
    BellmanFord(source);
    cout << "\nAll Shortest paths from source " << source << ":\n\n";
    for (int i = 1; i <= node; i++)
    {
        cout << "Path to vertex " << i << ": ";
        printPath(i);
        cout << "\nTotal Path cost: " << cost[i] << "\n\n";
    }
    return 0;
}

```