#include <iostream>

#include <vector>

#include <list>

#include <climits>

using namespace std;

const int MAX = 1e4 + 5;

const int INF = INT\_MAX;

vector<pair<int, int>> graph[MAX];

int dist[MAX];

list<int> bucket[MAX];

void dijkstra(int s, int n, int C) {

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

dist[i] = INF;

}

dist[s] = 0;

bucket[0].push\_back(s);

for (int d = 0; d <= C \* n; d++) {

while (!bucket[d].empty()) {

int u = bucket[d].front();

bucket[d].pop\_front();

if (dist[u] < d) continue;

for (auto &edge : graph[u]) {

int v = edge.first;

int w = edge.second;

if (dist[v] > dist[u] + w) {

dist[v] = dist[u] + w;

bucket[dist[v]].push\_back(v);

}

}

}

}

}

int main() {

int n, m, s, C;

cin >> n >> m >> s >> C;

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

int u, v, w;

cin >> u >> v >> w;

graph[u].push\_back({v, w});

}

dijkstra(s, n, C);

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

if (dist[i] == INF)

cout << "Không có đường đi từ " << s << " đến " << i << endl;

else

cout << "Khoảng cách ngắn nhất từ " << s << " đến " << i << " là: " << dist[i] << endl;

}

return 0;

}