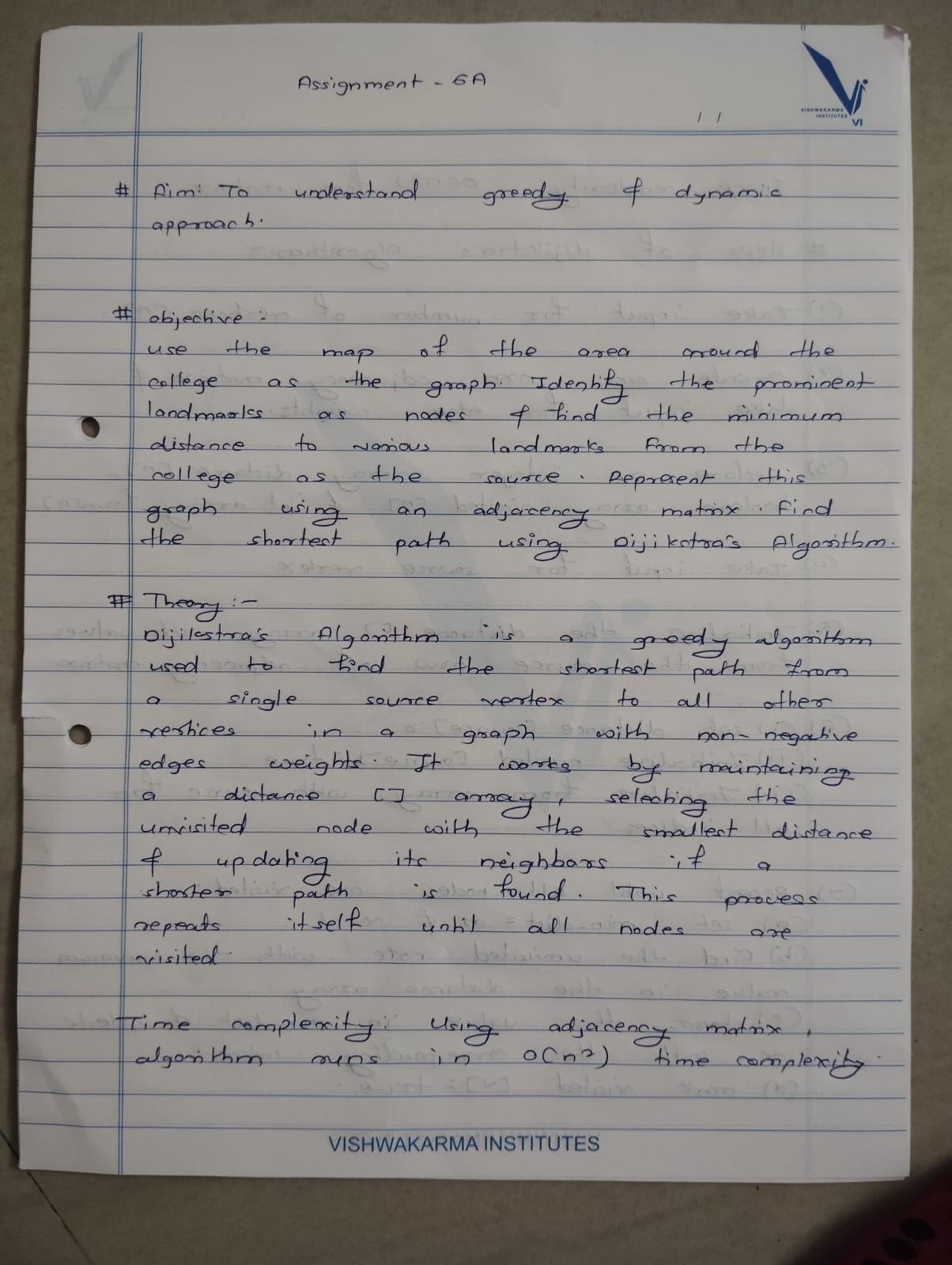
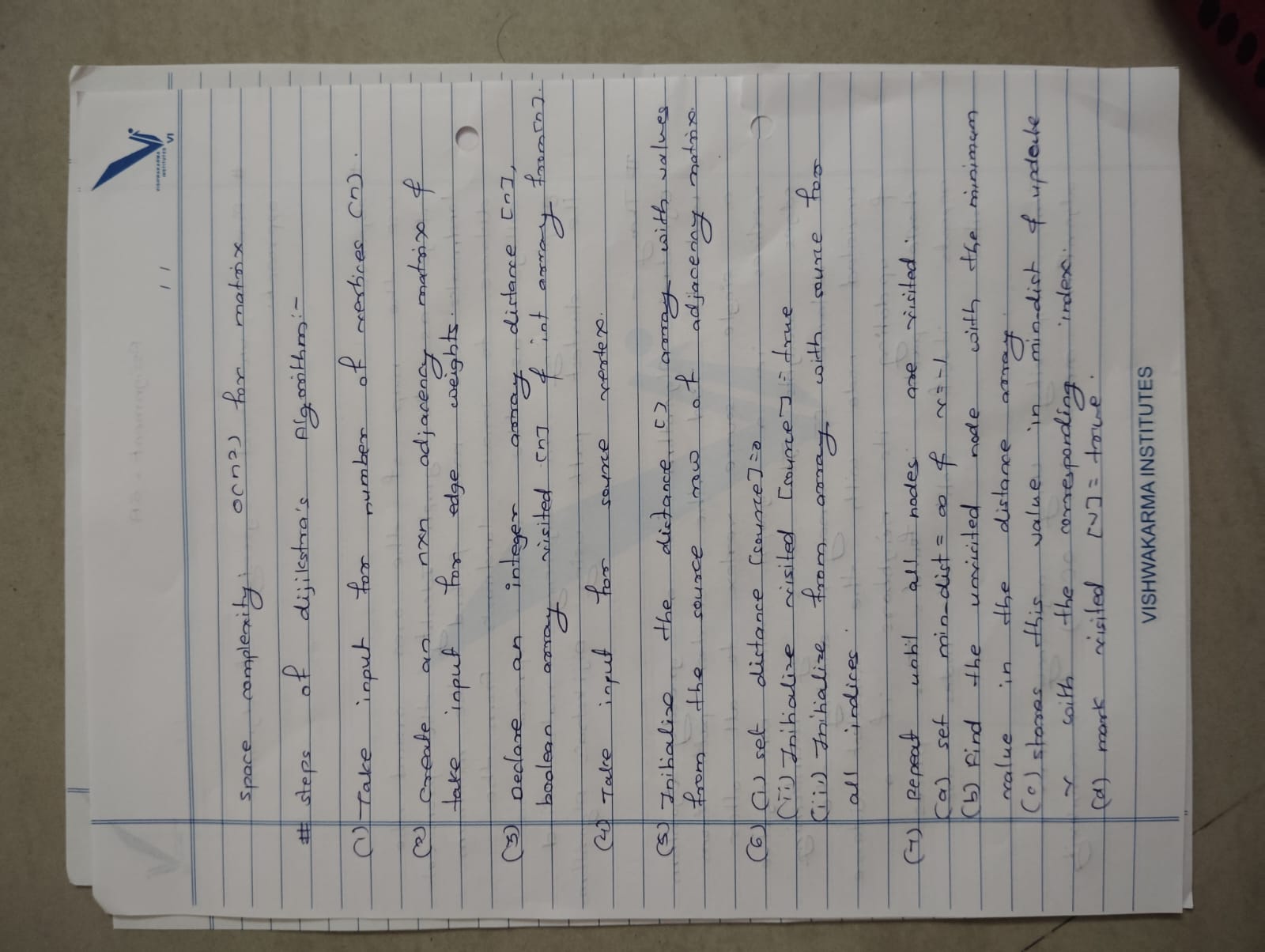
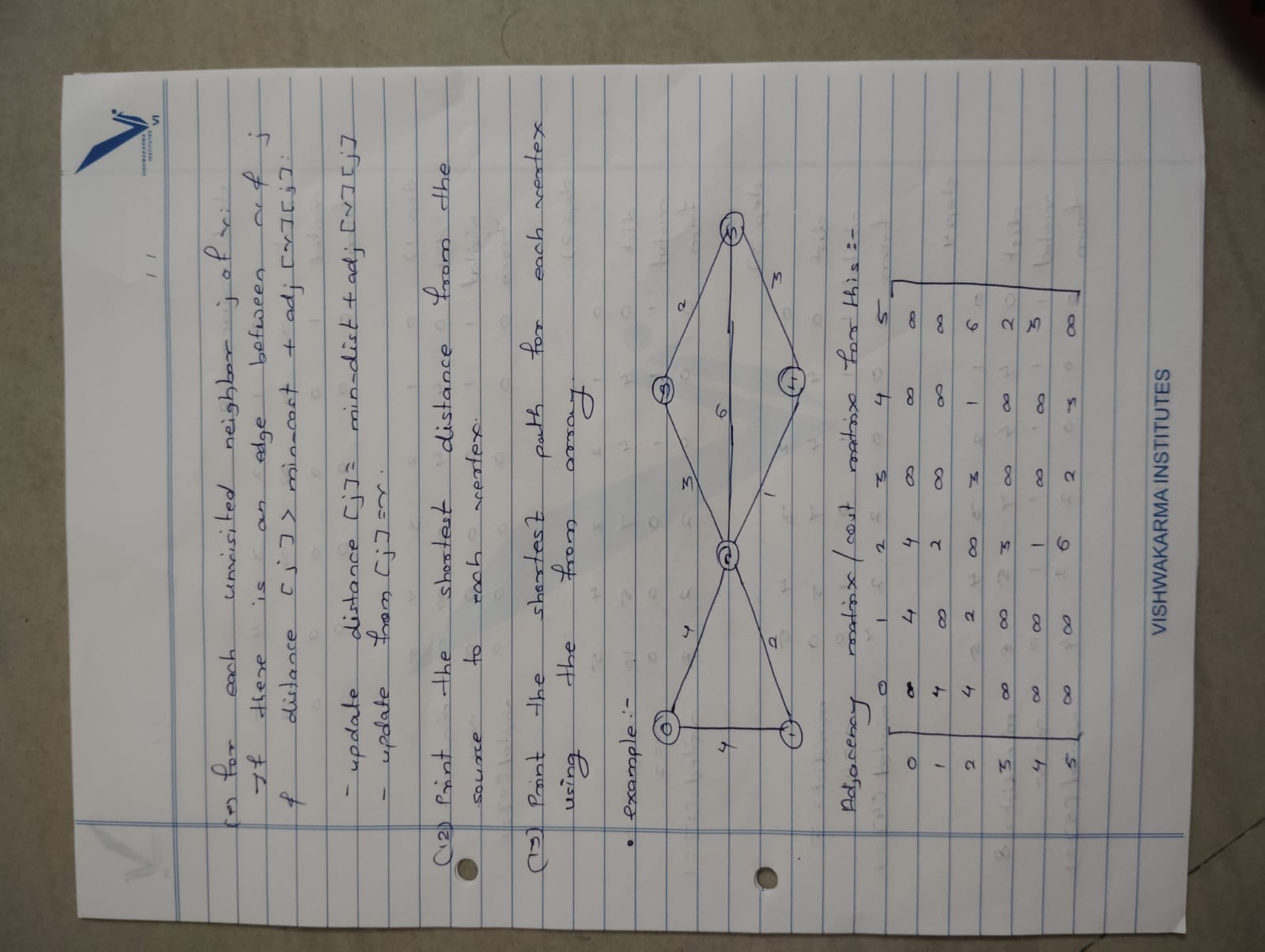
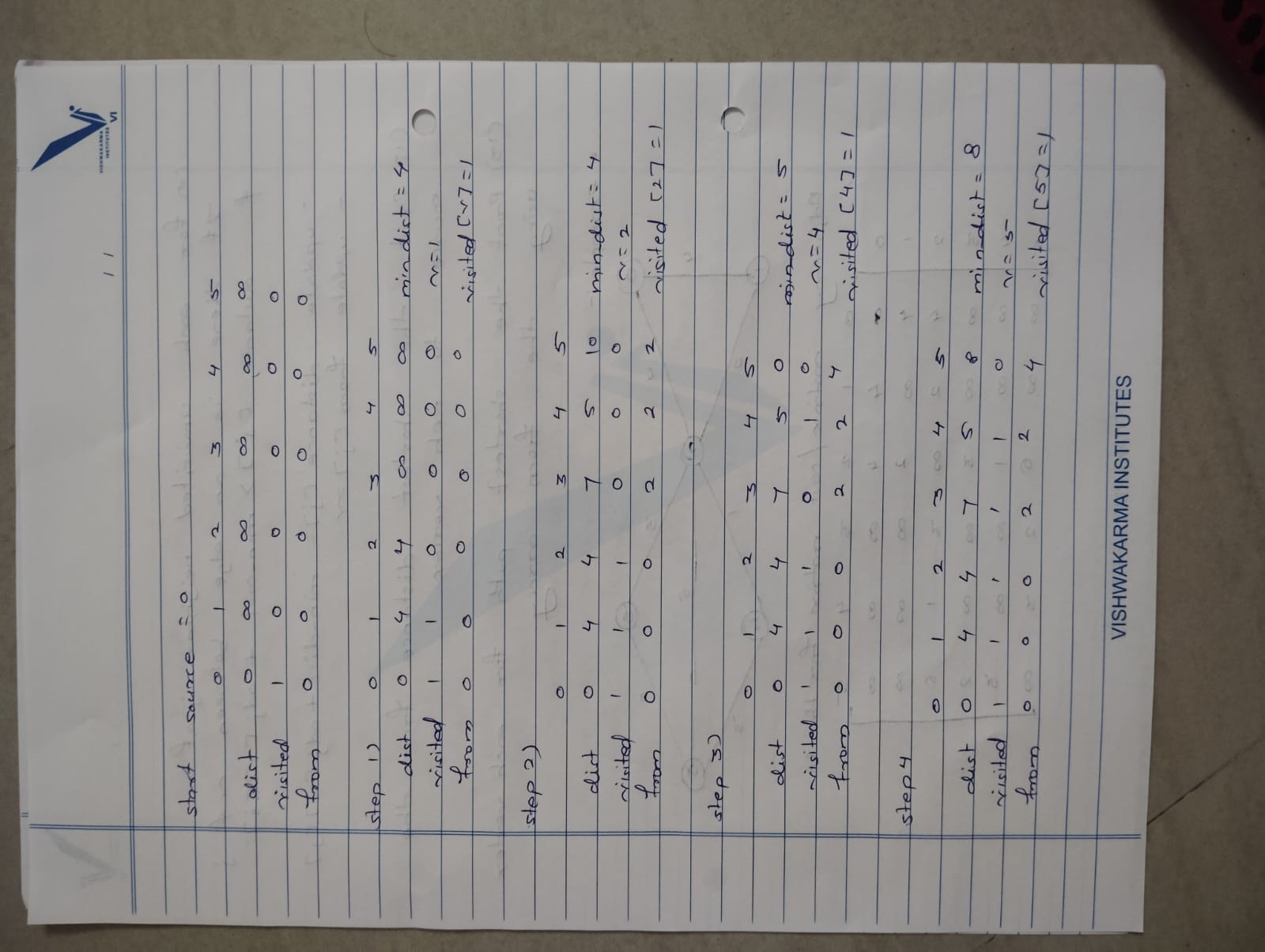
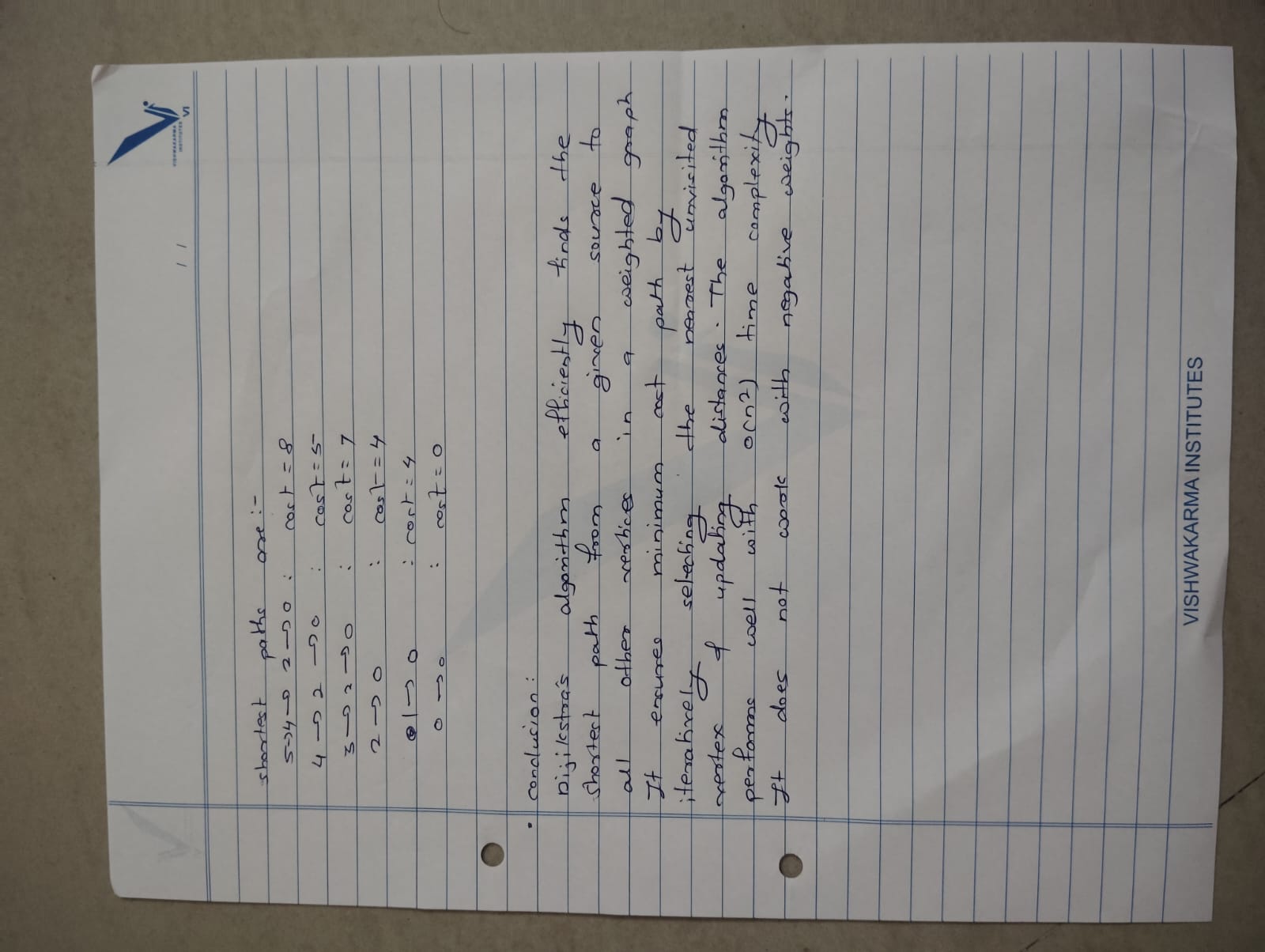
****

****

****

****

****

**Name:** Rohan Arun Nalawade

**Roll No:** 231012

**PRN:** 22310407

**SY IT A**

**Assignment 6(A)**

**Code Implementation:**

#include <iostream>

#include <vector>

#include <climits>

#include <queue>

using namespace std;

void dijkstra(const vector<vector<int>>& adjMatrix, int source, int numNodes) {

    vector<int> dist(numNodes, INT\_MAX);

    dist[source] = 0;

    priority\_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> pq;

    pq.push({0, source});

    while (!pq.empty()) {

        int u = pq.top().second;

        int d = pq.top().first;

        pq.pop();

        if (d > dist[u]) continue;

        for (int v = 0; v < numNodes; ++v) {

            if (adjMatrix[u][v] != INT\_MAX) {

                int newDist = dist[u] + adjMatrix[u][v];

                if (newDist < dist[v]) {

                    dist[v] = newDist;

                    pq.push({newDist, v});

                }

            }

        }

    }

    cout << "Shortest distances from node 0 (college):\n";

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

        if (dist[i] == INT\_MAX) {

            cout << "Node " << i << " is unreachable.\n";

        } else {

            cout << "Distance to node " << i << ": " << dist[i] << "\n";

        }

    }

}

int main() {

    int numNodes;

    cout << "Enter the number of landmarks (nodes): ";

    cin >> numNodes;

    vector<vector<int>> adjMatrix(numNodes, vector<int>(numNodes, INT\_MAX));

    cout << "Enter the adjacency matrix (use a large number or 'inf' for no direct edge):\n";

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

        for (int j = i; j < numNodes; ++j) {

            if (i == j) {

                adjMatrix[i][j] = 0;

            } else {

                cout << "Distance from node " << i << " to node " << j << ":";

                int distance;

                cin >> distance;

                if (distance >= 0) {

                    adjMatrix[i][j] = distance;

                    adjMatrix[j][i] = distance;

                }

            }

        }

    }

    int source = 0;

    dijkstra(adjMatrix, source, numNodes);

    return 0;

}

Output: