Assignment 12

Done By: Nayan Man Singh Pradhan

**Problem 12.1**

A friend wants help in implementing an algorithm for finding the shortest path between two nodes and in a directed graph (possibly containing negative edge weights). He/she proposes the following:

* Add a large constant to each weight such that all weights become positive
* Run Dijkstra’s algorithm for the shortest path from to

This method is not correct as it may not always work for all cases. A counterexample for when this method does not work:

Assume we have a start point and an end point with vertices/nodes.

**Path 1:**

Length of path 1:

**Path 2:**

Length of path 2:

We can see **that path 2 is the shorter path**. Now, we add the largest constant (in both cases ) in both the paths!

**Path 1:**

Length of path 1:

**Path 2:**

Length of path 2:

Now, in this case, when we add a positive large constant in both paths, **path 1 is the shorter path**. Therefore, we disprove the correctness of the given algorithm using a simple counter example!

**Problem 12.2**

Implemented in “OMP.cpp”. Execute make to run.

**Problem 12.3**

**a.**

The problem given to us can be represented as a graph problem.

1. Consider our board B.
2. B has coordinates B[x][y] that represents the position of the player.
3. Every coordinate B[x][y] (position) in the board is a node.
4. All edges are 1 as the distance moved is 1 (in all directions form the current position)
5. The vertex {0, 1, 2, .… , – 1}
6. Edges of vertex are neighboring nodes, i.e. up, down, left, right.