Test Case 1

Open test1.txt

Test Case 1 opens a file and inserts it into a graph. We then test different cases and see if the correct shortest path is found. Next, we will insert new nodes and edges and test more paths.

Operation Correct Shortest Path Found(y/n/no path)

Find shortest path

PS->SS1 y

PS->SS2 y

PS->SS3 y

PS->SS10 y

PS->SS11 y

PS->SS5 y

PS->SS4 y

SS1->SS0 no path

SS4->SS0 no path

SS10->SS9 y

SS3->SS7 y

Add new nodes

SS12->SS5(5)

SS13->SS11(10)

SS8->SS14(7)

SS14->SS15(10)

SS15->SS16(8)

SS9->SS16(14)

Find shortest path with new additions

SS13->SS11 y

SS16->SS0 y

SS15->SS16 y

SS0->SS16 y

Test Case 2

Open test2.txt

Test Case 2 opens a file and inserts its contents into a graph. We then test different paths to make sure the correct shortest path is found. Next we insert a new edge into the existing graph and test more to make sure the new edge gets put into the graph.

Operation Correct Shortest Path Found (y/n/no path)

Find shortest path

N0->N20 y

N0->N15 y

N0->N14 y

N14->N0 no path

N4->N0 y

N11->N20 y

N5->N16 y

N8->N19 y

Add new adjacency

N5->N20(25)

Find shortest path

N0->N20 y

Test Case 3

In Test Case 3 we manually insert vertices and edges to create a graph. We then test every possible path and see if it is a valid path or not to test if our program will catch invalid paths.

Insert Nodes (no weight needed)

N1->N2

N2->N4

N2->N5

N2->N7

N3->N6

N3->N2

N6->N4

N7->N6

Test All Possible Paths (1 if possible path, X if no possible path)

N1->N2 1

N1->N3 1

N1->N4 1

N1->N5 1

N1->N6 1

N1->N7 1

N2->N1 X

N2->N3 X

N2->N4 1

N2->N5 1

N2->N6 1\*

N2->N7 1

N3->N1 X

N3->N2 1

N3->N4 1

N3->N5 1

N3->N6 1

N3->N7 1

N4->N1 X

N4->N2 X

N4->N3 X

N4->N5 X

N4->N6 X

N4->N7 X

N5->N1 X

N5->N2 X

N5->N3 X

N5->N4 X

N5->N6 X

N5->N7 X

N6->N1 X

N6->N2 X

N6->N3 X

N6->N4 1

N6->N5 X

N6->N7 X

N7->N1 X

N7->N2 X

N7->N3 X

N7->N4 1

N7->N5 X

N7->N6 1

\*Problem found and fixed at N2->N6

Test Case 4

In test case 4 we test to see if the file I/O works properly.

I.

First we open test4.txt and display the graph to see if everything in the file was input into the graph. Next we will save the graph to the file. The files test4.txt and output.txt should look the same when we are finished.

Open File test4.txt

Display Graph (G) Correct Output(y/n)

y

Save Graph to File (S) output.txt Correct Output(y/n)

y

II.

In part II we open test4.txt to insert its contents into a graph. Next we manually add nodes and edges and save the graph to output2.txt and check the file for correctness.

Open test4.txt Correct Output(y/n

y

Add

N21->N22

N22->N23

Save Graph to File (S) output2.txt Correct Output(y/n)

y