Test Plan

MST Team 7: Luv Shah, Suzy Roche, Dom Pace, Marc Rosales

# Testing EdgeWeightedGraph Objects

1. Case 1: An Empty Graph

Should not throw an error. Check if the graph has an adjacency matrix that is an empty set “{}”, and 0 vertices, and weight of 0.

1. Case 2: A Negative number/Non-Numeric vertices graph

A graph cannot have negative vertices, must default to an empty graph, and a weight of 0. A negative graph defaults to a 0 weight graph.

1. Case 3: A non-empty graph

Given N vertices, with M weighted edges, create a graph with an adjacency of N x N dimensions, with 2\*M edges contained within the matrix.

The Weight of the graph is (E0.wieght + E1.weight + E2.weight + … + EM-1.weight)

# Testing Prim’s algorithm Minimum Spanning Tree Objects

1. Case 1: Empty Graph

Should not throw an error. The graph’s weight should be 0. Adjacency matrix is an empty set “{}” and 0 vertices.

1. Case 2: Non-Empty graph

Should match the correct pre-generated MST from an answer-key Prim’s algorithm output. The weight should equal the answer-key algorithm’s weight, which is equal to or less than the original edge weighted graph.

Background pattern

Description automatically generated Diagram

Description automatically generated

# Testing Kruskal’s algorithm Minimum Spanning Tree Objects

1. Case 1: Empty Graph

Should not throw an error. The graph’s weight should be 0. Adjacency matrix is an empty set “{}” and 0 vertices.

1. Case 2: Non-Empty graph

Should match the correct pre-generated MST from an answer-key Kruskal’s algorithm output. The weight should equal the answer-key algorithm’s weight, which is equal to or less than the original edge weighted graph.

Background pattern

Description automatically generated Diagram

Description automatically generated