Assignment 3

Team Members:

1. Dhrumil Amish Shah

Date:

2021-02-17

—

Subject:

Software Development Concepts

—

Professor:

Matthew Amy

Assignment 3: Background

➔ Working with graphs and minimum weight spanning trees using Kruskal's algorithm.

➔ Consider initial data points and group the data points so that all the points in the same cluster are similar in some way and points in different clusters are deemed sufficiently different from one another.

➔ Given a set of documents, we will compare each pair of documents and will create a number that reflects how similar the two documents are to one another.

Assumptions and Cases

1. I have coded in a way that calling function addEdge() as
   a. addEdge("A", "A", <any_weight>) or
   b. addEdge("A", null, <any_weight>) or
   c. addEdge(null, "A", <any_weight>)
   will work fine. Vertex "A" is not connected to the graph and <any_weight> will be ignored. Calling method clusterVertices() will always result "A" in a separate cluster.

2. I have coded in a way that calling addEdge() in below order
   a. addEdge("A", "A", <any_weight>) // <any_weight> ignored
   b. addEdge("B", "B", <any_weight>) // <any_weight> ignored
   c. addEdge("A", "B", <any_weight>)
   will work fine. It first adds the standalone vertex "A", then "B" and then it will join "A" and "B" with <any_weight>. Now when I run clusterVertices(), it will either merge "A" and "B" in the same cluster or both will be in a separate cluster based on the value of the tolerance.
   This is same as calling addEdge("A", "B", <any_weight>)

3. I have coded in a way that calling addEdge("A", "B", 3) first and then calling addEdge("A", "B", 10) will not update the edge weight between "A" and "B" from 3 to 10 but instead it will return false. (Indicates that an edge exists between "A" and "B").

4. I have not considered empty strings. (Document name cannot be empty). So, strings like "", " ", " ", "   ", etc. will return false.

Test Cases

⇨ Input Validation Cases

1. boolean addEdge(String vertex1, String vertex2, int weight)
    - Send vertex1 and vertex2 null.
        - Returns false. (Not allowed)
    - Send vertex1 and vertex2 empty string.
        - Returns false. (Not allowed)
    - Send negative weight.
        - Returns false. (Not allowed)

2. Set<Set<String>> clusterVertices(float tolerance)
    - Send negative tolerance.
        - Returns null. (Not allowed)

⇨ Boundary Cases

1. boolean addEdge(String vertex1, String vertex2, int weight)
    - Send valid vertex1 and vertex2 with weight 0.
        - Returns true. (Edge added)
    - Send valid vertex1 and vertex2 with weight 1.
        - Returns true. (Edge added)
    - Send valid vertex1 and vertex2 with weight as a max integer value.
        - Returns true. (Edge added)
    - Send one-character vertex1 and vertex2 with valid weight.
        - Returns true. (Edge added)
    - Send long string vertex1 and vertex2 with valid weight.
        - Returns true. (Edge added)

2. Set<Set<String>> clusterVertices(float tolerance)
    - Send 0 tolerance.
        - Returns set of clusters if no error otherwise null.
    - Send 1 tolerance.
        - Returns set of clusters if no error otherwise null.
    - Send maximum allowed tolerance.
        - Returns set of clusters if no error otherwise null.

⇨ Control Flow Cases

1. boolean addEdge(String vertex1, String vertex2, int weight)
    - Send valid vertex1 and null vertex2 with any weight.
        - Returns true. (Vertex added in a graph with no connected edge)
    - Send null vertex1 and valid vertex2 with any weight.
        - Returns true. (Vertex added in a graph with no connected edge)
    - Send equal vertex1 and vertex2 with any weight.
        - Returns true. (Vertex added in a graph with no connected edge)
    - Send a null graph.
        - Returns true. (Graph created with no edges)
    - Send a connected graph. (Example: Graph in Assignment 3 PDF).
        - Returns true. (Creates a connected graph)
    - Send separate vertices and connect with an edge.
        - Returns true. (Vertices connected)
    - Send duplicate edge. (Existing connected vertex1 and vertex2).
        - Returns false. (Duplicate edge not added)
    - Send duplicate vertex. (Existing vertex1 and vertex2 but not connected).
        - Returns false. (Duplicate vertex not added)
    - Send vertex1 connected to only vertex2.
        - Returns true. (Edge added)
    - Send case sensitive vertices.
        - Returns true. (Edge added)
    - Send valid non trimmed vertex1 and vertex2.
        - Returns true. (Edge added with vertex1 and vertex2 trimmed)

2. Set<Set<String>> clusterVertices(float tolerance)
    - Send valid tolerance on a graph with no edges (Null graph)
        - Returns set of clusters with each vertex in a separate cluster if no error otherwise null.
    - Call with valid tolerance value on a graph where all vertices are connected.
        - Returns set of clusters if no error otherwise null.
    - Call with valid tolerance value on a graph where some vertices are connected, and some are not.
        - Returns set of clusters if no error otherwise null.
    - Call on a graph where calculated tolerance is always greater than parameter tolerance

        - Returns set of clusters with each vertex in a separate cluster if no error otherwise null.
    - Call on a graph where calculated tolerance is always smaller or equal to parameter tolerance
        - Returns set of clusters with all vertices in one cluster if no error otherwise null.
    - Call with different tolerance value on the same graph.
        - Returns set of clusters if no error otherwise null.

⇨ Data Flow Cases

1. boolean addEdge(String vertex1, String vertex2, int weight)
    - Send a valid edge in an empty list.
        - Returns true. (Edge added)
    - Send a valid vertex in an empty list.
        - Returns true. (Vertex added)
    - Send a valid edge in a list with edges.
        - Returns true. (Edge added)
    - Send a valid vertex in a list with edges.
        - Returns true. (Edge added, self-edge with no weight)

2. Set<Set<String>> clusterVertices(float tolerance)
    - Call before calling addEdge().
        - Returns null. (Not allowed)
    - Call multiple times on the same graph.
        - Returns set of clusters if no error otherwise null.