Assignment 3

Problem 1:

Assumptions:

- Self-looping vertices are not added in the graph
    - So having just a single vertex without any edge is not allowed in the code
      (Though that functionality can be added)
- Two vertices with an edge completely separated from the main graph is possible with the code
- Negative Tolerance not allowed
    - Tolerance = 0 is allowed
- Vertex name like a document name works in the code
- Same vertex with a different weight will not be added as code checks whether an edge exists or not.
- Several edges with same weight are sorted lexicographically
- As it is an undirected weighted graph, the source & destination vertices are swapped lexicographically as well for any edge connecting them
- The final output containing the vertex clusters is not sorted
  (The functionality can be added if needed)
- Ignoring the round off error in the tolerance computation
- Vertices are Case Sensitive
- Merging the clusters if calculated tolerance is <= given tolerance

Test Cases:

Input Validation:

- Source string passed as Null to addEdge method
    - Returns false & edge is not created nor added
- Destination string passed as Null to addEdge method
    - Returns false & edge is not created nor added
- Extra leading & trailing spaces passed in Source/Destination string to addEdge method
    - Returns true & creates & adds an edge
      (trimming the extra leading & trailing spaces)
- Document name passed as string to Source & Destination to addEdge method
    - Returns true & creates & adds an edge
- Source & Destination string are the same
    - Returns false & edge is not created nor added
      (Assumption: self-looping is not allowed)
- Very long string passed as Source/Destination to addEdge method
    - Returns true & creates & adds an edge
- Very short string passed as Source/Destination to addEdge method
    - Returns true & creates & adds an edge
- 0 Tolerance passed to the clusterVertices method
    - Accepts the input & forms clusters & returns a 2d set

Note:
No input validation tests for addVertex method as vertices are added from the already validated allEdges arraylist

Boundary Test cases:

- Negative weight passed to addEdge method
    - Returns false & edge is not created nor added
- 0 weight passed to addEdge method
    - Returns false & edge is not created nor added
- Empty string passed as Source to addEdge method
    - Returns false & edge is not created nor added
- Empty string passed as Destination to addEdge method
    - Returns false & edge is not created nor added
- Weight value passed higher than the int data type range (both positive/negative)
    - Compile time error
- Negative Tolerance passed to the clusterVertices method
    - Returns null

(majorly covered in Input Validation part)

Control-Flow Test Cases:

- Adding an already present edge to addEdge method
    - Returns false & does not add the edge
- Adding an edge with same Source & Destination but different weight to addEdge method
    - Returns false & does not add the edge
- All edges with the same weight added
    - Clusters are formed
- All edges with different weight added
    - Clusters are formed

Data-Flow Test Cases:

- Instantiate Graph twice in a row
- Call addVertex method twice in a row
- Call addVertex method without adding any edges
- Call clusterVertices method without adding any edges
- Call clusterVertices method before calling addVertex method
- Calling custerVertices method twice in a row