CSE-2212: Design and Analysis of Algorithms-I Lab Practice Lab 8– December 23, 2024

Experiment: Implementing Algorithms in Java for Merge Sort, All Pair Shortest Path, and Articulation Points and Bridges

In this lab experiment, you will implement three algorithms—

1) Merge Sort
2) All Pair Shortest Path algorithm using Floyd-Warshall,
3) Finding articulation points and bridges in a graph.

Problem 1: Merge Sort
You will implement the Merge Sort algorithm, which recursively divides the array into smaller subarrays, sorts them, and merges them back together.

Method to Implement:

- mergeSort(int[] array, int left, int right): Recursively sorts the array.
- merge(int[] array, int left, int mid, int right): Merges two sorted subarrays.

Example Input:
Input Array: [12, 11, 13, 5, 6, 7]
Expected Output:
Sorted Array: [5, 6, 7, 11, 12, 13]

Problem 2: All Pair Shortest Path (Floyd-Warshall Algorithm)
You will implement the Floyd-Warshall algorithm to find the shortest path between all pairs of vertices in a graph.

Input:
The graph will be represented as an adjacency matrix. You will initialize the graph from an input file.
Method to Implement:

- floydWarshall(int[][] graph): Calculates the shortest path between all pairs of vertices.

Sample Input File:
4
0 3 INF 5
2 0 INF 4
INF 1 0 INF

INF INF 2 0

Expected Output:
0 3 7 5
2 0 6 4
3 1 0 5
5 3 2 0

Problem 3: Finding Articulation Points and Bridges

In this part, you will identify articulation points (vertices whose removal increases the number of connected components) and bridges (edges whose removal increases the number of connected components) in an undirected graph.

Methods to Implement:

- findArticulationPoints(): Finds all articulation points in the graph.
- findBridges(): Finds all bridges in the graph.

The graph will be initialized using an adjacency list representation from an input file.

Sample Input File:
5 5
0 1
0 2
1 2
1 3
3 4

Expected Output:

Articulation Points: [1, 3]
Bridges: [(1, 3), (3, 4)]

Main Class Example

```
public class LabExperiment {
   public static void main(String[] args) throws IOException {

  // Merge Sort
      int[] array = {12, 11, 13, 5, 6, 7};
      System.out.println("Input Array: " + Arrays.toString(array));
      MergeSort.mergeSort(array, 0, array.length - 1);
      System.out.println("Sorted Array: " + Arrays.toString(array));
```

```
    // Floyd-Warshall
        int[][] graph = {
            {0, 3, Integer.MAX_VALUE, 5},
            {2, 0, Integer.MAX_VALUE, 4},
            {Integer.MAX_VALUE, 1, 0, Integer.MAX_VALUE},
            {Integer.MAX_VALUE, Integer.MAX_VALUE, 2, 0}
        };
        System.out.println("Shortest distances:");
        FloydWarshall.floydWarshall(graph);

    // Articulation Points and Bridges
        Graph graphObj = new Graph("input.txt");
        System.out.println("Articulation Points:");
        graphObj.findArticulationPoints();
        System.out.println("Bridges:");
        graphObj.findBridges();
    }
}
```