Group Project Check Point 4 (Mile Stone 2):
Prim's and Kruskal's Algorithms
Code & Report (40% Group Project's Grade)

Due Date: Thursday November 21, 2019 by 11:59 pm (Mid night)

CLO: 9	SO: C

Important Note:

- It is a group project.
- Every group member will do the task which he has been assigned during the lecture.
- After combining the solution of all parts in main class, only member 1 of every group will upload the solution of the Project on Blackboard within the due date.
- Clearly write the name, ID, email and the name of part(s) which you have solved, as comment in the main class.
- Include screen shots of the output in your solution.
- Late submission or syntax errors, or the program not implementing the assigned tasks completely will result in 0 marks.

Team tasks:

Task 1 (Member 1 & 2): Implement **Prim's algorithm** with a **binary heap**.

Task 2 (Member 1 & 2): Implement **Kruskal's algorithm** with a **Union and Find** of edges (Weighted quick-union).

Task 3 (Member 3): **Compare the runnning time** of Prim's algorithm vs. Kruskal's algorithm (union by size) + **Junit test** + **generate Javadoc** file after documentation of the the project.

Task 4 (Member 3): Write **one page of your observations** for the running times of Prim's algorithm vs. Kruskal's algorithm.

Union-Find ADT

For this project, you will create a Java interface for the Union-Find ADT that supports

- makeset(x) creates a one-element set x.
- Find(x) returns a subset containing x.
- union(x, y) constructs the union of the disjoint subsets S_x and S_y containing x and y

Sample Output:

Suppose the graph's file contains the following input where the first line denotes number of vertices, second line denotes number of edges, and other lines denote source and target vertices along with the weight of each edge.

```
6
10
0 1 3
0 4 6
0 5 5
1 2 1
1 5 4
2 3 6
2 5 4
3 4 8
3 5 5
4 5 2
```

Figure 1: The graph's input file

```
Total weight of MST by Prim's algorithm: 15.0
The edges in the tree are:
Edge from 0 to 1 has weight 3.0
Edge from 1 to 2 has weight 1.0
Edge from 1 to 5 has weight 4.0
Edge from 5 to 4 has weight 2.0
Edge from 5 to 3 has weight 5.0
Running time of Prim's algorithm using Min-Heap as Min-Priority Queue
is 3152 Nano seconds
```

Figure 2: MSTs starts from vertex 0

Total weight of MST by Kruskal's algorithm: 15.0
The edges in the tree are:
Edge from 1 to 2 has weight 1.0
Edge from 4 to 5 has weight 2.0
Edge from 0 to 1 has weight 3.0
Edge from 1 to 5 has weight 4.0
Edge from 3 to 5 has weight 5.0

```
Running Time of Kruskal's algorithm using Union-Find approach is 3457
Nano seconds.
```

Figure 3: MSTs starts from vertex 0

Note:

Only **Member 1** of every group will **upload** the solution on Blackboard always.

Group project (Checkpoint 4) has been uploaded in "Group Project" option. It is a group project. Its due date is **Thursday November 21, 2019 till 11:59 PM**. You can also upload the solution by **Friday November 22, 2019 till 11:59 PM with 25% deduction of marks and by Saturday November 23, 2019 till 11:59 PM with 50% deduction of marks**. **After this date, no submission will be accepted.**