Group Project: Checkpoint 3 (Milestone 1): Dijkstra's Algorithm

Due Date: Saturday November 02, 2019 till 11:59 PM

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.
- **Add screen shots of output in your submission.**
- Clearly write the name, ID, email and the name of part(s) which you have solved, as comment in the main class.
- Late submission or syntax errors, or the program not implementing the assigned tasks completelywill result in 0 marks.

Team tasks:

- **Member 1**: Implement Min-Priority Queue using unordered array
- **Member 2**: Implement Min-Priority Queue using binary heap.
- **Member 3**: Allow the user to find all-pairs shortest paths using Dijkstra's algorithm. Compare the running time of the Dijkstra's algorithm using Min-Priority queue using unordered array vs. Min-Priority queue using binary heap + JUnit test + generate Javadoc file after documentation of the project.

Note: Implement validation when it is possible such as If graph contains any negative weight edge, the program should display message Dijkstra’s algorithm does not work for this graph, and other validation checks etc.

Sample Output:

Suppose the graph's file contains the following input where the first line checks whether the graph is weighted or not, second line contains the number of vertices, second line contains number of edges, and other lines contain edges with source, target and weight of the edges respectively.

```
WEIGHTED
5
7
0 3 7
1 0 3
1 2 4
2 4 6
3 1 2
3 2 5
4 3 4
QUIT
```

Figure 1:

After you read the graph as adjacency matrix you will probably have the following output:

```
Weight  Matrix:

     0     1     2     3     4

0    0     0     0     7     0
1    3     0     4     0     0
2    0     0     0     0     6
3    0     2     5     0     0
4    0     0     0     4     0

Adjacent Vertices of every vertex:

VERTEX: 0 {a} - VISIT: false - ADJACENCY: 3
VERTEX: 1 {b} - VISIT: false - ADJACENCY: 0,2
VERTEX: 2 {c} - VISIT: false - ADJACENCY: 4
VERTEX: 3 {d} - VISIT: false - ADJACENCY: 1,2
VERTEX: 4 {e} - VISIT: false - ADJACENCY: 3

Starting from source vertex 0:
DFS Traversal: 0 ,3 ,1 ,2 ,4
BFS Traversal: 0 ,3 ,1 ,2 ,4
The graph is connected


    # of vertices is: 5, # of edges is: 7
  0:    0-3 7
  1:    1-0 3    1-2 4
  2:    2-4 6
  3:    3-1 2    3-2 5
  4:    4-3 4

Enter Source vertex: 4

Dijkstra using priority queue:
Shortest paths from vertex 4 are:
A path from 4 to 0: 4 3 1 0  (Length: 9.0)
A path from 4 to 1: 4 3 1  (Length: 6.0)
A path from 4 to 2: 4 3 2  (Length: 9.0)
A path from 4 to 3: 4 3  (Length: 4.0)
A path from 4 to 4: 4  (Length: 0.0)


Dijkstra using min heap:
Shortest paths from vertex 4 are:
A path from 4 to 0: 4 3 1 0  (Length: 9.0)
```

```
A path from 4 to 1: 4 3 1  (Length: 6.0)
A path from 4 to 2: 4 3 2  (Length: 9.0)
A path from 4 to 3: 4 3  (Length: 4.0)
A path from 4 to 4: 4  (Length: 0.0)


Comparison Of the running time :
    Running time of dijkstra using priority queue is: 3599939 nano seconds
    Running time of dijkstra using min Heap is: 1195418 nano seconds
    Running time of dijkstra using min Heap is better
```

Figure 2: Finding shortest paths from source vertex 4

Priority Queue ADT:

For this project, you will create a Java interface for the Priority Queue ADT that supports
insert(v, p) : insert vertex v with priority p.
deleteMin() : delete and return the vertex with the minimum priority.
decreaseKey(v, p) : lower the priority for vertex v to p.

Note:

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

Group project (Checkpoint 3) has been uploaded in “Group Project” option. It is a group project. Its due date is **Saturday November 02, 2019 till 11:59 PM**. You can also upload the solution by **Sunday November 03, 2019 till 11:59 PM with 25% deduction of marks and by Monday November 04, 2019 till 11:59 PM with 50% deduction of marks**. **After this date, no submission will be accepted.**