**Notes:**

* Problems 1 and 2 are due by 9:50 AM, November 15, 2017.
* You are required to turn in a written report (Word or PDF file) for the homework part (problems 3 and 4) of the lab and upload implementations to canvas. These are due by 8:00 AM, November 29, 2017).

**Objectives:**

* Implement weighted graph and Prim’s Algorithm for minimum spanning trees.

**Problems:**

1. Implement a weighted graph class from the Graph.java used in Lab 11 and 12. Graph.java uses integer value for storing an edge. Instead of using integer value for storing edges, create an “Edge” class that holds information of edge. Edge class has following attributes (not limited to)
   1. Vertex v1, v2
   2. int edgeID
   3. int edgeWeight

Sample Output : [1] 2(1.0) 4(6.0)

Change the adjacency list to hold the vertex and edge weight.

1. Write a driver program, which reads input files mediumGraph.txt, LargeGraph.txt and XtraLargeGraph.txt (downloadable from Canvas) and display the weighted graphs by printing adjacency list.
2. Implement Prim’s algorithm (provided in the next page). Use your pre class assignment to check whether the output of your code is correct or not.
3. Write a driver program, which reads input files mediumGraph.txt, LargeGraph.txt and XtraLargeGraph.txt (downloadable from Canvas) and run Prim’s algorithm on each of them to find the minimum spanning tree within these graphs. Record the times required for each of these graphs

NB: for the following pseudo code, you need to use min-heap queue. In Java, you can use PriorityQueue API.

|  |  |
| --- | --- |
|  |  |

