```
CS 6301.002.  Implementation of advanced data structures and algorithms
Spring 2016;  Thu, Feb 18.
Long Project 2: Minimum spanning trees

Ver 1.0: Initial description (Feb 18: 1:00 PM).
Max excellence credits: 1.0 (only for level 2).
Due: 11:59 PM, Sun, Mar 13 (1st deadline), Sun, Mar 27 (2nd deadline).
```

Code base

Java library, java examples discussed in class by instructor. Do not use code from outside sources.

Project Description

Write algorithms for finding minimum spanning trees. You must complete either level one (Prim1, Prim2, Kruskal), or level two (Edmonds' branching algorithm [MST in directed graphs]).

Level 1

No excellence credits will be awarded for submitting level 1. You need not submit this part if you are submitting code for level 2. An undirected graph is given as input, in the format expected by readGraph method. Implement 3 algorithms: Prim1 (priority queue of edges; use Java's priority queues), Prim2 (priority queue of vertices, using indexed heaps), and Kruskal's algorithm. Provide 3 driver programs for the 3 algorithms. You can reuse SP0pq code for this project. Output just the weight of the minimum spanning tree.

Level 2

Implement the algorithm discussed in class for finding minimum spanning trees in directed graphs. The input format is the same as level 1 (use readGraph method to create the graph). Assume that the root is vertex 1.

Output specification

In the first line of the output, print the weight of a minimum spanning tree. Note that though the edge weights are ints, the weight of an MST may not fit in an int. If |V| is less than or equal to 50, in the next lines, output the edges of an actual MST. Order the edges of the MST by their "To" nodes. If |V| is more than 50, then output just the weight of the MST.

```
Sample input:
5 7
1 5 8
1 4 7
1 3 6
4 3 3
3 5 6
5 3 2
5 2 1

Output:
17
(5,2)
(4,3)
(1,4)
(3,5)
```