CSE 2331
(Kruskal's Algorithm Programming Assignment)
Autumn 2018

THIS PROGRAMMING ASSIGNMENT SHOULD BE DONE INDIVIDUALLY. You may discuss the algorithm with other students BUT DO NOT LOOK AT OR COPY ANYONE ELSE'S CODE.

In this homework you will write a programs to read an edge-weighted directed graph from a file, graphX, and report the MINIMUM SPANNING FOREST of graphX calculated using Kruskal's algorithm, using a list-based UNION-FIND data structure with wheighted union to test for cycles.

The graph file format is as follows:

- The first line of the contains the number of vertices in the graph (n).
- Each successive line contains three integers separated by a coma and a colon respectively. The first integer is the starting vertex of the edge, the second one is its destination and the third one is the cost or weight associated with that edge. For example, the line `0,3:6` would represent the edge from vertex v_0 to vertex v_3 with a cost of 6 (and of course, could only exist in a graph with $n \geq 4$

The algorithm would output the list of edges and the their cost in the order they are added by the algorithm. For example, if the lowest cost edge in the graph goes from vertex $v1$ to vertex v_3 and has the cost of 7, we would expect the first line of the output to be `3,0:7`.

You will have to implement Kruskal's algorithm using a list based Union-Find data structure that uses union wheighted by the length of the sets. A version of Kruskal's algoritm provided below, but if you need further details, look into Chapter 23.2 in the book.

```
   procedure KruskalsMSF(G)
 1 foreach vertex v_i ∈ V(G) do
 2 |  F.MakeSet(v_i) ;                          /* F is a union-Find data structure */
 3 end
 4 Q.Initialize () ;                           /* Q is a min-priority queue */
 5 foreach edge (v_i, v_j) ∈ E(G) do
 6 |  Q.Insert((v_i, v_j, w))
 7 end
 8 while Q.IsNotEmpty() and F.Size() > 1 do
 9 |  e → Q.RemoveMin() ;
10 |  if F.Find(e.source) ≠ F.Find(e.destination) then
11 |  |  report(e);
12 |  |  F.Union(e.source,e.destination);
13 |  end
14 end
```

Program specification:

- Your program should be written in Java, C, C++, or Python3.
- To use java in stdlinux, type "subscribe" and then select JDK-CURRENT. To compile sumRange.java in stdlinux, type "javac sumRange.java". (Use the javac command to compile other files/classes.) To run the program sumRange, type "java sumRange dataX rangeX".
- To use Python in stdlinux, type "subscribe" and then select JDK-Python-3.
- Name your programs `KruskalsMSF`.
- Your program has one command line argument, graphX representing an edge wheighted directed graph and a vertex in the graph repectively. A user runs your java program on file graphX, by typing the following command:

  ```
  java KruskalsMSF graphX
  ```

- Sample graphX files will be provided in Carmen.

- All coding for this lab is to be done individually. You may discuss this lab with other students but DO NOT LOOK AT OR COPY anyone else's code.
- Check that your program runs and compiles on stdlinux. The grader will compile and test the program on stdlinux. Programs which do not compile on stdlinux will receive 0 points.

In the dropbox on carmen submit:

1. A plain text file named README giving instructions on how to compile and run your program. (If your program needs to link libraries, be sure to include these in the instructions.)
2. Source code (NOT object code) for your program.

Do NOT submit object code and certainly DO NOT submit any data files. The grader will compile and run your program.