Homework 11: MSTs

Due: See T-Square

Important

There are general homework guidelines you must always follow. If you fail to follow any of the following guidelines you risk receiving a **0** for the entire assignment.

1. All submitted code must compile under **JDK 8**. This includes unused code, so don't submit extra files that don't compile. (Java is backwards compatabile so if it compiles under JDK 7 it *should* compile under JDK 8.)
2. Do not include any package declarations in your classes.
3. Do not change any existing class headers, constructors, or method signatures.
4. Do not add additional public methods when implementing an interface.
5. Do not use anything that would trivialize the assignment. (e.g. don't import/use `java.util.LinkedList` for a Linked List assignment. Ask if you are unsure.)
6. You must submit your source code, the `.java` files, not the compiled `.class` files.
7. After you submit your files redownload them and run them to make sure they are what you intended to submit. You are responsible if you submit the wrong files.

Minimum Spanning Trees

You will be implementing two algorithms to compute Minimum Spanning Trees: Prim's algorithm and Kruskals algorithm. You may import any standard Java data structure for either algoirthm (think carefully about which ones are needed). You are also required to implement a Disjoint Sets data structure for use in Kruskal's algorithm

Style and Formatting

It is important that your code is not only functional but is also written clearly and with good style. We will be checking your code against a style checker that we are providing. It is located in resources along with instructions on how to use it. We will take off a point for every style error that occurs. If you feel like what you wrote is in accordance with good style but still sets off the style checker please email Jonathan Jemson (jonathanjemson@gatech.edu) with the subject header of "CheckStyle XML". **Please make sure that you are using the CheckStyle XML file named CS1332-checkstyle-v2.xml. All homework assignments will use this CheckStyle.**

Javadocs

Javadoc any helper methods you create in a style similar to the Javadocs for the methods in the interface.

Provided

The following files have been provided to you:

1. `MST.java` This is class you will code. Each method has javadoc descriptions of what it should do. Do not add any non-constant global variables. **Do not add any new public methods.**
2. `DisjointSetsInterface.java` This is the interface for the DisjointSets ADT. Each method has javadoc descriptions of what it should do. **Do not modify this file.**

3. DisjointSets.java This is class you will code. This should implement the interface above. Do not add any non-constant global variables. **Do not add any new public methods.**
4. Vertex.java A class that represents a vertex in the graph used in the algorithms. **Do not modify this file.**
5. Edge.java A class that represents an edge in the graph used in the algorithms. **Do not modify this file.**
6. Graph.java A class that represents the graph used in the algorithms. **Do not modify this file.**
7. StudentTests.java This is the test class that contains a set of tests covering the basic cases for Disjoint Sets and MSTs. It is not intended to be exhaustive nor guarantee any type of grade. **Write your own tests to ensure you cover all edge cases.**

Deliverables

You must submit all of the following files. Please make sure the filename matches the filenames below.

1. MST.java
2. DisjointSets.java

You may attach each file individually, or submit them in a zip archive.

Be sure you receive the confirmation email from T-Square, and then download your uploaded files to a new folder, copy over the interface, recompile, and run. It is your responsibility to re-test your submission and discover editing oddities, upload issues, etc.