

CS146: Quiz 12  
Due Tuesday, May 2, at 7:00AM  
10 points

You will implement two methods in the Graph class. **To get started, import the starter file, Graph.java into the mst package you create in a new Java Project.** This graph class looks a little different than both Quiz 10 and Homework 6. I've set it up to make the implementation of Kruskal's algorithm easier. For this quiz you will implement Kruskal's algorithm to compute the minimum spanning tree of the graph and a method that determines if the graph is connected. Note that edges in this graph are undirected. **However you cannot create two edge objects for each undirected edge. Instead you should think about how you can use one edge object to represent the undirected edge.** Please do not change any of the method signatures in the class, but you can add any helper methods you deem necessary. Implement the methods described below. You are free to test your code however you prefer. These methods should be completed individually using any IDE you are comfortable with. You are free to use the textbook, slides, class notes, and the [Java API Documentation](#), but **DO NOT** consult any other resources.

#### Vertex Class (DO NOT EDIT)

The vertex class holds information about the vertices in the graph. It implements the Comparable Interface so you can store the Vertex objects in a TreeSet. Note that each vertex value will be unique.

#### Edge Class (DO NOT EDIT)

This class represents the edges in the graph. Note that we should think of the edge object as representing an undirected edge.

#### Disjoint Set Class (DO NOT EDIT)

This class implements the Union-Find algorithm used in Kruskal's MST algorithm.

```
public boolean connected()
```

This method returns true if the Graph object is connected. Remember this is an undirected graph. An undirected graph is connected if there is a path between any 2 vertices. (Hint: think about how we can use depth first search or breadth first search to help us solve this problem).

```
public Graph kruskalMST()
```

This method returns the minimum spanning tree as a new Graph object. This minimum spanning tree should have the same vertices as the original graph object, but only a subset of the edges. We introduced the pseudocode for this algorithm during class on Thursday 4/27. You should use the Disjoint Set class to determine if an edge is safe to add to the MST.

## Submission

Please create a jar or zip file of your project. It should include the java file (Graph.java) and submit it on Canvas. If your file includes class files or other extraneous files outside of the package folder and java files, you will receive an automatic one point deduction.