Last name _____

First name _____

LARSON—MATH 356–CLASSROOM WORKSHEET 27

1. (a) Start Chrome browser.
   (b) Go to `http://cocalc.com` and "Sign In".
   (c) Click project **Math 356**.
   (d) Click "New", call it **c27**, then click "Sage Worksheet".

   A **spanning tree** of a connected graph G is a subgraph of G that contains all the vertices of G which is a tree.

   For a connected weighted graph, a **minimum weight spanning tree** is one whose edges have a minimum weight sum for all possible spanning trees.

2. Let *pete* be the Petersen graph. Evaluate `pete=graphs.PetersenGraph()` and "show" the graph. Draw it below.

3. Find a spanning tree for pete (by hand).

4. Argue that every connected graph has a spanning tree.

5. Argue that every connected weighted graph has a minimum weight spanning tree.

```
def random_integer_edge_weights(g,n): #weights will be in the interval [1..n]
    E = g.edges()
    for e in E:
        random_weight = randint(1,n)
        g.add_edge(e[0],e[1], random_weight)
    return g
```

6. In order to use Kruskal's algorithm, we will need some examples of weighted graphs. The above code will choose random integers from 1 to n for each edge. Let `pete1=random_integer_edge_weights(pete,10).show(edge_labels=True)`. "Show" it and draw what you get. (This will be different for each student).

7. Try to find a minimum weight spanning tree of pete1 by hand.

8. Evaluate the code for Kruskal's algorithm on p.65 of our book. Try it on pete1. Draw what you get.

9. How can you use your Kruskal's code to find a spanning tree of an unweighted graph?

10. Implement your own algorithm for finding a spanning tree of a graph.