

Last name _____

First name _____

LARSON—MATH 356—CLASSROOM WORKSHEET 29

1. (a) Start Chrome browser.
(b) Go to `http://cocalc.com` and “Sign In”.
(c) Click project **Math 356**.
(d) Click “New”, call it **c29**, 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.

We need some examples of weighted graphs. Last week we coded the following function that puts random integer weights on the edges of a given graph g :

```
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
```

3. 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).

4. List the edges of your *pete1* graph from smallest weight to largest weight.

5. Now use Kruskal’s algorithm (by hand) to find a minimum weight spanning tree of *pete1* by hand, and draw what you get. What is the total weight of your spanning tree?

6. Evaluate the code for Kruskal's algorithm on p.65 of our book. Try it on `pete1`. Draw what you get. What is the total weight of this spanning tree?
7. How can you use your Kruskal's code to find a spanning tree of an unweighted graph?
8. Our next task is to implement an algorithm that takes a connected graph g and a vertex v as input and outputs a list of levels sets $[L_0(v), L_1(v), \dots, L_k(v)]$. where $L_k(v)$ is the last non-empty level set. Describe an algorithm that does this.
9. What should your function return for the Petersen graph and vertex 0.
10. What auxiliary functions will we need to implement your ideas?
11. Start hacking away at a function `level_sets(g,v)` that implements your algorithm.