Last name _____

First name _____

LARSON—MATH 356–SAGE WORKSHEET 10
Prufer Codes!

Reminders

1. Read ahead in our textbook. Up currently is Prufer codes of trees and Cayley's Theorem. Up next is Euler circuits (Sec. 4.1)

Coding Algorithms

1. Log in to your Sage/CoCalc account.
   (a) Start the Chrome browser.
   (b) Go to `http://cocalc.com` and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click "New", call it **s10**, then click "Sage Worksheet".
   (e) For each problem number, label it in the Sage cell where the work is. So for Problem 1, the first line of the cell should be `#Problem 1`.
   (f) When you are finished with the worksheet, click "make pdf", email me the pdf (at `clarson@vcu.edu`, with a header that says **Math 356 s10 worksheet attached**).

Saving and Re-using Code

We've coded several graphs now, and have added code for functions of graph invariants and auxiliary functions and stored them in "graphs.sage". I pushed my updated version to your Handouts folder. Either copy that file to your Home directory—or add the new stuff to your own "graphs.sage" file. We'll need those functions.

2. I've updated the copy of "graphs.sage" in your Handouts folder to include what we've added in class. *Copy* the current version from Handouts to your Home directory.
3. *Load* your copy of "graphs.sage". Run: `load('graphs.sage')`.
4. Generate and display a `random_weighted_graph` with 5 vertices.

Recursive Functions & a Tree theorem

A graph G is a tree if and only if it has a degree-1 (leaf, pendant) vertex v and $G - v$ is a tree. We can use this idea to write a *recursive* function for testing whether a graph is a tree: see if it has a leaf, peel it off, and repeat for the remaining graph.

A **recursive** function is a function that calls itself. It must always have a *base case* so that the recursion eventually stops.

5. Here is an example of a recursive definition of the *factorial* function. The base case here is the case where the input is 0 or 1.

```
def factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n*factorial(n-1)
```

   Now try `factorial(0)`, `factorial(1)`, `factorial(2)`, `factorial(3)`, and `factorial(10)`.

6. The Fibonacci numbers are: 1,1,2,3,5,8,13,21..., where the next number is the sum of the previous two numbers. Write a recursive function function that computes the n^{th} Fibonacci number.

7. Write a function `recursive_is_tree` to test whether a graph is a tree by seeing if it has a leaf, peeling it off, and repeating for the remaining graph. You'll need a base case. What will it be?

Prufer Codes

8. How can we find a *cut vertex* in a graph?

   We will need the following two functions for our Prufer code construction algorithm.

9. Write a function `find_leafs(T)` to find all leafs in a tree T.

10. Write a function `find_cut_vertices(T)` to find all cut vertices in a tree T.

11. Write a function `find_prufer_vertex(T)` to find the label of the cut vertex incident to a leaf with smallest label in a tree T (assuming T's vertex labels go from 0 to $\nu-1$, for coding purposes, any list of linearly ordered labels will do).

    The **Prufer code** of a labeled tree T with labels $t_1, t_1, \ldots, t_\nu$ (from 0 to $\nu - 1$, for coding purposes, any list of linearly ordered labels will do) is a list $s_1, \ldots, s_{\nu-2}$ where s_1 is the label of the cut vertex v_1 adjacent to the leaf w_1 with the smallest label in tree $T = T_1$; s_2 is is the label of the cut vertex v_2 adjacent to the leaf w_2 with the smallest label in the tree $T_2 = T_1 - w_1$ (formed by deleting leaf w_1 and its incident edge from T_1; etc.

    In general s_i is is the label of the cut vertex v_i adjacent to the leaf w_i with the smallest label in the tree $T_i = T_{i-1} - w_{i-1}$ (formed by deleting leaf w_{i-1} and its incident edge from T_{i-1}.

12. Write a function `prufer_code(T)` that takes a labeled tree T (with labels from 0 to $\nu - 1$, for coding purposes, any list of linearly ordered labels will do) as input and outputs the Prufer Code of that tree.