Last name ______________________

First name ______________________

LARSON—MATH 255–SAGE WORKSHEET 06
Independent Sets—Independence Number

1. Log in to your Sage/Cocalc account.
   (a) Start Chrome browser.
   (b) Go to `http://cocalc.com`
   (c) Click "Sign In".
   (d) Click project **Classroom Worksheets**.
   (e) Click "New", call it **s06**, then click "Sage Worksheet".

   Sage has a built-in `Graph` class. A **graph** is a mathematical object consisting of *vertices* and *edges*.

   Evaluate `g=graphs.CompleteGraph(5)`. g is a Graph object. ("g" is our choice for the name of the graph. You could have used a different name). Many methods have already been defined for the Graph class. `show()` is one of these. Try `g.show()`. To see the methods available to a graph g, just type `g.` followed by the TAB key.

   The *independence number* of a graph is the largest number of vertices in the graph that have no edges between them. So if `g=graphs.CompleteGraph(5)`, its independence number is 1.

2. Find the independence number for `p3=graphs.PathGraph(3)` (by hand). Use the `show()` method to help you visualize.

3. Find the independence number for `c5=graphs.CompleteGraph(5)` (by hand). Use `c5.show()` to help you visualize.

4. Find the independence number for `p=graphs.PetersenGraph()`.

   Now we will write an algorithm to find the independence number. We will write it as an ordinary function (rather than as a Graph method). The first thing we need to be able to do is to test whether the vertices S from a graph g are *independent*. This means there are no edges between the vertices in S. So we need to *search* through the edges of g.

5. Evaluate `p.edges(labels=False)` to get a list of the edges of the Petersen graph. Evaluate `E=p.edges(labels=False)` to give this list the name E. Evaluate `(0,1) in E` to test if $(0,1)$ is the collection of edges E. Now check if $(0,2)$ is an edge.

6. Now we'll test every pair i and j from a set of vertices S to check if S is independent. If S is independent then the test for (i, j) will be false for each possible pair.

```
def is_independent(g, S):
    E=g.edges(labels=False)
    for i in S:
        for j in S:
            if (i,j) in E:
                return False
    return True
```

7. Use `is_independent()` to test if the sets $[1, 2, 3]$ and $[1, 2]$ are independent in $p3$. Find the largest independent set S you can find in the Petersen graph. Test if it is independent.

   The naive (and inefficient) way to find a largest independent set in a graph is to test every subset of vertices, check if it is independent, and then keep track of the largest one you've seen up to that point.

8. Let's try this by hand first for a smallish example. List all 8 subsets of the point set $\{0, 1, 2\}$ of $p3$.

9. Now, for each of those 8 subsets, check (by hand) if it is a independent set.

10. Of these 8 sets which is the largest independent set?

11. We'll use the subsets generator of a list to run through the subsets. Let `V=p3.vertices()`. Evaluate V to see what you have. lets see how it works. Let `L=subsets(V)`. Now try:

```
for S in L:
    print S
```

12. Now we'll write our first function to find a maximum independent set of a graph.

```
def maximum_independent_set(g):
    independent = []
    L=subsets(g.vertices())
    for S in L:
        if is_independent(g,S)==True:
            if len(S) > len(independent):
                independent = S
    return independent
```

13. Use this function to find a maximum independent set of the Petersen graph. Check that it agrees with your hand calculation.