LARSON—MATH 255–CLASSROOM WORKSHEET 31
Stable Sets & Stability Number

1. (a) Start the Chrome browser.
   (b) Go to `http://cocalc.com`
   (c) Login using **your VCU email address** .
   (d) Click on our class Project.
   (e) Click "New", then "Worksheets", then call it **c31**.
   (f) For each problem number, label it in the Sage cell where the work is. So for Problem 2, the first line of the cell should be `#Problem 2`.

   The *stability number* α of a graph is the largest number of vertices in the graph that have no edges between them.

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

2. How many subsets will we create for a graph that has n vertices? For each vertex v we will generate every possible set containing v and every possible set not containing v. So that will be 2 possibilities for each of the n vertices. That's 2^n subsets. (For our earlier example, thats $2^3 = 8$ subsets).

3. Here is a first function to find a maximum stable set of a graph. The auxiliary function is a test for whether or not a specific set S of vertices is stable: If S is stable then the test for (i, j) will not be an edge (will not be in the list of edges) of graph g for each possible pair i, j of vertices.

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

def naive_maximum_stable_set(g):
    stable = []
    L=subsets(g.vertices())
    for S in L:
        if is_stable(g,S)==True:
            if len(S) > len(stable):
                stable = S
    return stable
```

4. Use this function to find a maximum stable set of the Petersen graph.

   Then *prove* that the stability number for the Petersen graph is 4. To do this you need two things: (1) a stable set with 4 vertices, and (2) an argument that no stable set can have more than 4 vertices.

The next big idea for finding a maximum stable set was due to Tarjan and Trojanowski in the 1970s: they noted that each vertex v of a graph is either in a maximum stable set *or* it is not. And, if v is in a maximum stable set then none of the vertices it is touching (that it is *adjacent* to is), called the *neighbors* of v, can be in that set.

5. Find the neighbors of vertex 9 in the Petersen graph.

6. So our problem of finding a maximum stable set in a graph can be *reduced* to the problem of finding a maximum stable set in two smaller subgraphs: (1) the graph formed by removing vertex v and (2) the graph formed by removing v and its neighbors. In this case, we assume that v is in the maximum stable set.

7. To simplify things in the future, we wrote a function to remove a vertex and its neighbors from a graph and produce a new graph with v and its neighbors removed.

```
def remove_vertex_and_neighbors(g,v):
    S2=g.vertices()
    S2.remove(v)
    for w in g.neighbors(v):
        S2.remove(w)
    return g.subgraph(S2)
```

8. Type in the following (recursive!) code. Do you see how it implements the Tarjan-Trojanowski idea? Notive how this code builds up from pieces coded earlier. We're writing sophisticated code here! There are a couple of subtle issues that I will tell you more about in class.

```
def tt_maximum_stable_set(g, StableSet):
    V = g.vertices()
    if V == []:
        return StableSet
    v = V.pop()
    S1 = V
    S2 = remove_vertex_and_neighbors(g,v)
    g1 = g.subgraph(S1)
    g2 = g.subgraph(S2)
    Max1 = tt_maximum_stable_set(g1, StableSet)
    Max2 = tt_maximum_stable_set(g2, StableSet+[v])
    if len(Max1) > len(Max2):
        return Max1
    else:
        return Max2
```

Try `tt_maximum_stable_set(g, [])`. Now **test** this function with a variety of other graphs where you `know` the answer. Does it work?

Getting your classwork recorded

When you are done, before you leave class...

(a) Click the "Make pdf" (Adobe symbol) icon and make a pdf of this worksheet. (If Cocalc hangs, click the printer icon, then "Open", then print or make a pdf using your browser).

(b) Send me an email with an informative header like "Math 255—c31 worksheet attached" (so that it will be properly recorded).