Last name _____

First name _____

LARSON—MATH 556–SAGE WORKSHEET 08
Linear Programming Bound for the Stability Number of a Graph

1. Log in to your Sage Cloud account.

   (a) Start Firefox or Chrome browser.

   (b) Go to `http://cloud.sagemath.com`

   (c) Click "Sign In".

   (d) Click project **Classroom Worksheets**.

   (e) Click "New", call it **s08**, then click "Sage Worksheet".

   The *independence number* (or *stability number*) of a graph is the largest number of points in the graph that have no edges between them.

   For small graphs we can calculate the stability number by solving an integer program. Let x_1, x_2 ans

   maximize: $x_1 + x_2 + x_3$

$$\begin{array}{rrcccccc} & x_1 & + & x_2 & & & \leq & 1 \\ \text{subject to:} & x_1 & & & + & x_3 & \leq & 1 \\ & & & x_2 & + & x_3 & \leq & 1 \end{array}$$

   $x_1, x_2, x_3 \in \{0, 1\}$.

   Because the variables are restricted to integers this is an *Integer Programming* (IP) problem. If we allow the variables to be real numbers instead of integers (that is, to *relax* the IP) we get a linear program (LP). The relaxation of the stable set IP gives an optimum value that is necessarily an upper bound for the independence number.

   Here's how we can write this in Sage in a general way.

```
def fractional_alpha(g):

    p = MixedIntegerLinearProgram(maximization=True)
    x = p.new_variable(nonnegative=True)
    p.set_objective(sum(x[v] for v in g.vertices()))

    for v in g.vertices():
        p.add_constraint(x[v], max=1)

    for (u,v) in g.edge_iterator(labels=False):
        p.add_constraint(x[u] + x[v], max=1)

    return p.solve()
```

2. Type in the above definition and evaluate.

3. The graph represented by the in the above IP is the complete graph $k3$ on three vertices. Evaluate the following to get an upper bound for the independence number of $k3$:

```
k3=graphs.CompleteGraph(3)
fractional_alpha(k3)
```

4. Now find the LP upper bound for the Petersen graph Evaluate:

```
g=graphs.PetersenGraph()
g.show()
fractional_alpha(g)
```

5. It is possible to find LP bounds for graphs where calculating the true independence number is impossible. Let g be a random graph on 100 vertices. Evaluate: `g=graphs.RandomGNP(100,.5)`? What do you expect the upper bound to be. Now calculate.