1. Log in to your Sage Cloud account.
   (a) Start the Chrome browser.
   (b) Go to http://cloud.sagemath.com and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click “New”, call it c38, then click “Sage Worksheet”.

Mathematical objects not only have associated *invariants* but they also have *properties*, some condition $P$ that is either true or false for each member of the class. An example of a matrix property is being square: any given matrix is square or it is not. An example of a graph property is being Hamiltonian: any given graph is either Hamiltonian or it is not. An example of an integer property is being prime: any given integer is prime or it is not.

Properties define classes: the objects that have property $P$ define a class—the class of objects having that property. Similarly, classes define properties: the class $P$ defines the property of being a member of that class—any given object has the property of membership or non-membership in class $P$.

The CONJECTURING program can be used to generate conjectures about relationships between properties of objects (property-relation conjectures). The conjectures have the form $P \subseteq Q$, that is, “if a an object is in class $P$ then is in class $Q$”. Given a property of interest $P$, CONJECTURING can generate necessary conditions $Q$ for having property $P$ (these are also called *upper bounds* for $P$).

CONJECTURING can also generate *sufficient* condition conjectures: given a property $Q$ it will generate $P$’s that (might) imple membership in class $Q$ (these are also called *lower bounds* for having property $Q$).

2. We now investigate property-relation conjectures for graphs. Recall that a graph is *Hamiltonian* if there is a path that goes through all the vertices, without ever repeating a vertex, and returns to the starting vertex. It is a very difficult computational problem to determine if a graph is Hamiltonian—hence necessary and sufficient conditions for Hamiltonicity are of interest—and widely studied (more than a hundred are known).

Check if the following graphs are Hamiltonian: $k3, k4, k5, p3$.

3. Several graph properties are built-in to Sage. We can also define out own. Here’s one we previously discussed and used:

```python
def is_dirac(g):
    n = g.order()
    return min_degree(g) >= n/2
```
Check if the following graphs are Dirac: $k_3$, $k_4$, $k_5$, $p_3$.

4. Let's generate lower bound conjectures for a graph to be Hamiltonian. Investigate!

```python
load("conjecturing.py")

k3=graphs.CompleteGraph(3)
k4=graphs.CompleteGraph(4)
k5=graphs.CompleteGraph(5)
p3 = graphs.PathGraph(3)

def is_dirac(g):
    n = g.order()
    return min_degree(g) >= n/2

properties = [Graph.is_hamiltonian, is_dirac, Graph.is_tree,
              Graph.is_planar, Graph.is_connected]
objects = [k3,k4,k5,p3]
propertyBasedConjecture(objects, properties,
                         properties.index(Graph.is_hamiltonian))
```

5. Now let's generate upper bound conjectures for a graph to be Hamiltonian. Investigate!

```python
load("conjecturing.py")

k3=graphs.CompleteGraph(3)
k4=graphs.CompleteGraph(4)
k5=graphs.CompleteGraph(5)
p3 = graphs.PathGraph(3)

def is_dirac(g):
    n = g.order()
    return min_degree(g) >= n/2

properties = [Graph.is_hamiltonian, is_dirac, Graph.is_tree,
              Graph.is_planar, Graph.is_connected]
objects = [k3,k4,k5,p3]
propertyBasedConjecture(objects, properties,
                         properties.index(Graph.is_hamiltonian), sufficient = False)
```