COMP 4030/6030 Spring 2018. Assignment 4
Due: Feb 27, 2018 (before class starts)

1. (10 points) Use the geometric sum to find the answer for $1 + 6 + 6^2 + 6^3 + \ldots + 6^{31}$.
2. (30 points) Use repeated substitution to find the running time of T(n) = 4n + T(n-1). Assume T(1) = 1.
3. (30 points) Use repeated substitution to find the running time of T(n) = 4n + T(n/2). Assume T(1) = 1. If you can do problem 2 and know how to use substitution, you should be able to do this problem.

Programming questions:

4. (20 points) Given a random pairing of boys and girls, we know that it can have many unstable pairs. In this problem, you will design an algorithm that improves upon a random pairing by removing unstable pairs, one by one. There is no warranty that you can actually decrease the number of unstable pairs, because when you "stabilize" an unstable pair, you might introduce another unstable pair. However, intuitively, removing unstable pairs, one at a time, should improve upon a random pairing of boys and girls.

   Your Python function should take as input a list of pairs of boys and girls. Strategically, your function iterates through each pair in this list, check if it is unstable, and if so, fix it. The API looks like this:

```
def  improve_stability( pairs ):
        # Strategy: iterate through each pair and "fix" it if it is unstable.
        # This function should call another function, find_unstable_pair, see below.
        # If the current pair is unstable, you fix it by swapping boys and girls in the
        # two pairs.  You should remove these two pairs from the list and add two
        # new pairs to the list.
        # Python lists have methods append (for adding) and remove (for deletion).
```

We discussed and implemented in class a function that detect unstable pairs. You can modify this function to return None if the pair is unstable or in case it is unstable a pair that makes it unstable. For example, if ('John', 'Rose') is stable, then your function, find_unstable_pair(('John', 'Rose')) returns None. If, however, ('John', 'Rose') is unstable, then find_unstable_pair(('John', 'Rose')) returns the pair, say ('Joe', 'Mary'), that makes it unstable. A minor modification of is_stable (which we implemented in class) should work.

This is a decrease-and-conquer strategy at work. We attempt to remove unstable pairs one by one.

In this algorithm, you go through the list of pairs only once, trying to fix unstable pairs. It is possible that after you finish going through the list, there are still unstable pairs. For now, this is expected.

5. (10 points)Compare your algorithm to the random arrangement. The relationship module, as you should know, has a random_arrangement function that returns a list of random pairs. You can count the number of unstable pairs in the random pairing, to the number of unstable pairs returned from your algorithm.

   Write a function called compare_to_random that iterates 100 times, and prints out the average number of unstable pairs of the random pairing and of your improved pairing.

Plagiarism Policy:
You can discuss how to solve the problems with your classmates, but the solution must be your own. Using other people's solution will result in a zero for the assignment and possible additional penalties.

Submission:
Put your name as part of the file name and upload your submission to eCourseware Dropbox.