Lab Interlude: Asymptotic Analysis
CSCI 2101 – Fall 2021

Due (Section A): Sunday, October 3, 11:59 pm **(firm; no flex days may be used)**
Due (Section B): Tuesday, October 5, 11:59 pm **(firm; no flex days may be used)**
Collaboration Policy: Level 1
Group Policy: Individual

This lab interlude will explore the fundamentals of asymptotic analysis and Big-O notation. You may either type or neatly handwrite your solutions. All solutions must be uploaded electronically to Blackboard (e.g., by scanning or taking a clear picture).

1. Show that 2^{n+1} is $O(2^n)$ by finding c and n_0 to satisfy the big-O requirement. Explain why your chosen values work.

2. Show that 2^{2n} is *not* $O(2^n)$ by showing that it is not possible to find c and n_0 to satisfy the big-O requirement. Note that $2^{2n} = (2^n)^2$.

3. Give a big-O characterization (and brief justification) of the running time, in terms of n, of each of the following five loops. Think in terms of the number of loop iterations that will be required. Note that the sum of the arithmetic sequence $1, 2, 3, \cdots, k$ is $\frac{k}{2}(1+k)$.

   Algorithm Loop1 (n):
   $s \leftarrow 0$
   for $i \leftarrow 1$ to n **do**
   $s \leftarrow s + i$

   Algorithm Loop2 (n):
   $p \leftarrow 1$
   for $i \leftarrow 1$ to $2n$ **do**
   $p \leftarrow p * i$

   Algorithm Loop3 (n):
   $p \leftarrow 1$
   for $i \leftarrow 1$ to n^2 **do**
   $p \leftarrow p * i$

   Algorithm Loop4 (n):
   $s \leftarrow 0$
   for $i \leftarrow 1$ to $2n$ **do**
   for $j \leftarrow 1$ to i **do**
   $s \leftarrow s + i$

   Algorithm Loop5 (n):
   $s \leftarrow 0$
   for $i \leftarrow 1$ to n^2 **do**
   for $j \leftarrow 1$ to i **do**
   $s \leftarrow s + i$

4. Given a `SimpleArrayList` of initial size n, give a big-O characterization (and justification) of the running time of the following Java function, in terms of n:

```
public void doubleList(SimpleArrayList<Integer> myList) {
    int size = myList.size();
    for (int i = 0; i < size; i++) {
        int pos = rand.nextInt(myList.size()); // rand is a Random object
        myList.add(pos, i);
    }
}
```

   Would your answer change if the the fourth line instead read "`int pos = myList.size();`"? If so, what would be the new running time and why?

5. Explain whether the following statement is true or false:

   "If choosing between an $O(n \log n)$ algorithm and an $O(n^2)$ algorithm to solve a problem on a specific input, it is always better to use the $O(n \log n)$ algorithm."

   Assume that the two algorithms use equivalent space and that the algorithms are already implemented (so you do not need to worry about the difficulty of implementation, for instance).