SAN FRANCISCO STATE UNIVERSITY
Computer Science Department

CSC510 Section 04 – Analysis of Algorithms
Algorithm Challenge 1: Complexity Functions

Instructor: Jose Ortiz

08/04/2020

Full Name: _____

Student ID: _____

Assignment Instructions. Must read!

Note: Failure to follow the following instructions in detail will impact your grade negatively.

1. This algorithm challenge is worth 10%, and will be graded using a grading point scale where the maximum possible grade is 100 points. For instance, if your grade in this assignment is 85/100, then this is equivalent to 0.85*10%=8.5% of 10%

2. The deadline of this assignment will be announced by the instructor in class.

3. Each section of this algorithm challenge is worth 25 points

4. Take into account that in this type of assignments, I am more interested in the way you approach the problem rather than your final solution.

Problem Statement

1. Create an optimized function "print_s(n,s)" that prints the given argument **s** (representing a string) **k** times. **k** represents the number of iterations of '**for j**' for each iteration of **i** based on the increments $i = i + 1$ and $j = j * 2$

   (a) Initial conditions: $i = 1$, $j = 1$, $i <= n$, and $j <= i$
   (b) input as arguments in the function: n (an integer representing the size of the input), and s (the string)
   (c) output: print s k times
   (d) example: n=5, s="hello CSC510-01 class", s will be printed 11 times

Your work here

1. Describe the algorithm to solve the problem. Use n=5 as your base example, and then based on your solution, define a general algorithm for all the values of n. Finally, state the complexity of your algorithm as (1) a function of T(n), and (2) time complexity with big O notation

2. Write the pseudocode to that defines the algorithm in part (1) for $T(n)$

3. Provide an optimization for the pseudocode in part (2). Note that there is always a way to optimize your algorithm. I want you to think hard about this.

4. Create/implement the method based on your work above and provide several unit test for your (optimized)solution