SAN FRANCISCO STATE UNIVERSITY
Computer Science Department

CSC510 – Analysis of Algorithms
Algorithm Challenge 1: Step Counting

Instructor: Jose Ortiz

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 9%, and will be graded using a grading point scale where the maximum possible grade is 9 points
2. Handwriting work or screenshots of your work are not allowed. In addition, all the pseudocode done in this algorithm challenge must be done using LaTeX. Students who fail to meet this policy won't get credit for their work. Note that for pseudocode, I only want to see the compiled PDF psudocode, instead of the code to create the pseudocode.
3. Each section of this algorithm challenge is worth 2.25 points
4. Take into account that in this type of assignments, I am more interested in all the different approaches you take to solve the problem rather than on the final solution.

Problem Statement

1. Create the pseudocode for the function "print_s(n,s)" that prints the given argument **s** (representing a string) **k** times. **k** represents the number of iterations of the inner loop. The print statement that prints **s** is in the inner loop of the function
   (a) Initial conditions: $i = 1$, $j = 1$, $i <= n$, and $j <= i$
   (b) i and j increments: $i = i + 1$ and $j = j * 2$
   (c) input as arguments in the function: n (an integer representing the size of the input), and s (the string)
   (d) output: print s k times
   (e) example: n=5, s="hello CSC510-01 class", given these conditions, the function "print_s(n,s)" will print **s** 11 times

Your work begins here

1. Describe the algorithm to solve the problem. Note that you must show all your work (e.g table of steps like I did in my lectures)

   (a) Use n=5 as your base example to create the algorithm

   (b) Refine your algorithm from (a) to cover all the cases for any given n size

2. Write the pseudocode (code is not allowed here) that represents your algorithm in part (1.b). **This needs to be done in LaTeX. Otherwise, students won't get credit for this problem**

3. Based on your pseudocode from part 2, compute the complexity function of your algorithm $T(n)$, and its worst case time complexity using $Big\ O$ notation. **Show all your work, including detailed computation of the summations to get credit for this problem**

4. Try to optimize (improve) the time complexity of your algorithm and if so, write (using LaTeX) the pseudocode for the new optimization. I want you to think hard about this. Also, if you cannot find an optimization, explain why.