Name: Daniel Kim
ID: 102353420

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Exam 2 – S18 **Spring 2020, CU-Boulder**

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit (30 min for 1x, 37.5 min for 1.5x, 45 min for 2x). Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

YOU MUST SIGN THE HONOR PLEDGE. Your quiz will otherwise not be graded. Honor Pledge: On my honor, I have not used any outside resources (other than my notes and book), nor have I given any help to anyone completing this assignment.

Your Name: Daniel Kim _____

Standard 18. Suppose we have n stairs to climb. You may choose to jump up either $1, 2$, or 3 stairs. Your goal is to count the number of ways to climb the stairs. Note that your starting position is on the ground floor and not on the first stair.

Is there a clear recursive structure in the problem that would be useful in designing an effective dynamic programming algorithm? That is, is dynamic programming a useful algorithmic technique for this problem? Clearly justify your answer.

From the prompt, we have 3 options to jump which are 1, 2, or 3 stairs which also implies (n - 1), (n - 2), and (n - 3), n for number of stairs to climb. In other words, let's say that we are at the (n - 1)th stairs, this means that we can get up to n by jumping 1 more step. 2 more jumps for (n - 2)th stairs, 3 more jumps for (n - 3)th stairs. To reach up to n stairs, we could be at either (n - 1)th, (n - 2)th or (n - 3)th stairs. To add on, within the original problem, there can be 3 sub problems because we have 3 base cases: to get up to 1 stairs there is only one way, to get up to 2 stairs there are 2 ways, to get up to 3 stairs we have 4 ways. $(1+1+1), (1+2), (2+1), (3)$. So we get a recursive structure in this problem.

Let's say that T(n) is the number of ways to reach n stairs,
T(n) = T(n - 1) + T(n - 2) + T (n - 3)
AND 2^{n-1} with n greater or equal to 1 and n less than or equal to 3

Therefore, dynamic programming is a useful algorithmic technique for this problem.