Module 3: Advanced Written Problems

1. Suppose you are given a row of n doors, each of which can either be locked or unlocked. Your goal is to have exactly S secured doors. The issue is that not all locked doors are secure. A door is secure if and only if:

   1. This door is locked AND
   2. This door is the first door (i.e., the left most door) OR The door to this one's left is also locked

   Develop an algorithm that, given the number of doors n and a value S, counts the total number of unique ways the doors can be locked to achieve EXACTLY S secure doors.

2. Consider a situation in which you have n skiers with heights $p_1, p_2, ..., p_n$, and n pairs of skis with length $s_1, s_2, ..., s_n$. The problem is to assign each person a pair of skis so that the overall difference between the height of a skier and the length of their skis is minimized. More precisely, if $\alpha(i)$ returns the index of the skis assigned to person p_i, we want to minimize:

$$\frac{1}{n}\sum_{i=1}^{n}(|p_i - s_{\alpha(i)}|)$$

   Consider two greedy algorithms that may solve this problem:

   1. Find the skier and skis whose height difference is minimized. Assign those skis to that skier and repeat.
   2. Pair the smallest skier with the smallest skis, second smallest skier to second shortest skis, etc. until all pairings are complete.

   One of these greedy algorithms is correct. You must do three things:

   1. Prove this problem has optimal substructure.
   2. Claim which of the two algorithms is correct.
   3. Show the greedy choice property of your chosen algorithm by using an exchange argument.

3. Suppose you are given a rectangular board of size 4 by $w \geq 1$ (the units here don't matter). You are also given an endless bag of dominoes, each of size 2 by 1. Write a program that accepts the integer w as parameter, and returns the total number of unique ways the 4 by w board can be fully tiled. When we say *fully tiled* here, we mean that every square on the board is covered by dominoes. State the runtime of your algorithm.

For example, if the input given is $w = 2$, then the solution is 5, the following image shows all of the combinations of fully tiled boards of width $w = 2$:

HINT: To solve this program, start by thinking of all the unique ending patterns a tiling can have. They are in the image below. Instead of using a recurrence to count all the tilings, count the number of tilings that have EACH individual ending pattern. Then think about ways you can take solutions with one ending pattern, to produce tilings of one width longer by making a few changes. The total number of tilings will be the sum of the number of tilings of the individual patterns. Good luck!