Homework 10

Discrete Mathematics

In general, late homework will not be accepted. However, you are allowed to turn in **up to two late homework assignments**. Unless you have made arrangements in advance with me, homework turned in after class will be considered late.

Complete the following problems. Unless explicitly stated otherwise, you are expected to justify your answers. In many problems this means that you should use words to describe what you are doing and why. In other problems, simply providing sufficient arithmetic may be sufficient. If a problem asks you to count something, please $\boxed{\text{box}}$ your final answer.

1. Prove that the number of binary strings of length n that never have two consecutive 1's is the Fibonacci number f_{n+2}. See our textbook for the definition of the Fibonacci numbers.
2. How many compositions of n have all their parts greater than 1, except possibly the last entry. For example, for $n = 9$, $(3,4,2)$ and $(3,3,2,1)$ are acceptable, but not $(3,3,1,2)$.
3. Solve the following recurrence relations.
   (a) $a_n = a_{n-1} + 11$ with initial condition $a_1 = 7$.
   (b) $a_n = -5a_{n-1}$ with initial condition $a_1 = 40$.
4. Let s_n denote the number of subsets of $[n]$ that contain no two consecutive elements. For example, $\{1,3,7\}$ is such a set, but $\{1,2,7\}$ is not since 1 and 2 are consecutive. Determine s_n in terms of a recurrence, a closed form, or by describing a bijection to a set we already know how to count.
5. A **Dyck path** of length $2n$ is a lattice path from $(0,0)$ to (n,n) that takes n steps East from (i,j) to $(i+1,j)$ and n steps North from (i,j) to $(i,j+1)$ such that all points on the path satisfy $i \leq j$. This sounds more complicated that it really is. You can think of a Dyck path as one of our paths to get coffee that starts at $(0,0)$ and ends at (n,n) but never drops below the line $y = x$. Let $\mathrm{Dyck}(n)$ denote set of all Dyck paths of length $2n$ and let $d_n := |\mathrm{Dyck}(n)|$. We define $d_0 := 1$ for convenience. *Important:* Unfortunately, we also used d_n to denote the number of derangements of n. This problem is about Dyck path, not derangements. Compute d_1, d_2, d_3, and d_4 via brute force.