Problem Set 7 - Math

<u>Question C</u>

When approaching this question, it was pretty easy to identify that it was a DP question, and from this, a matrices question - to speed up the DP. The question was just 1. Trying to figure out what that recurrence was, and then 2. Trying to understand the matrix application to recurrences from the lecture.

The recurrence wasn't too hard to figure out. If we take d[i] as the number of ways that we can split/not split i gems, such that the space occupied by them is i, then in order to link it to previous states, we just need to consider the case where the i'th gem is split, and where it isn't. When the i'th gem isn't split, we want to get the number of ways to fill d[i-1] space. If the gem is split, we want to get the number of ways to fill d[i-M] space.

Thus, the final recurrence would be d[i] = d[i-1] + d[i-M]. The matrix was the more fun part, I honestly did not understand the concept of matrices' application to recurrence before this one, and once it clicked, it was great. My transition matrix was built M x M, with 1 0 0 0 … 0 0 1 in the top row, and the following rows standard. Following the example from freddy the frog, we could just use d{1}, d{0}, d{-1} etc. as the base case, and then raise the matrix to the power of n instead of n - m.

Finally, after the matrix exponentiation, the answer would be stored in the 0, 0 value due to the fact that our base case was written to be (1, 0, 0, 0 … 0, 0). The logic follows from the Freddy Frog example in the lecture.