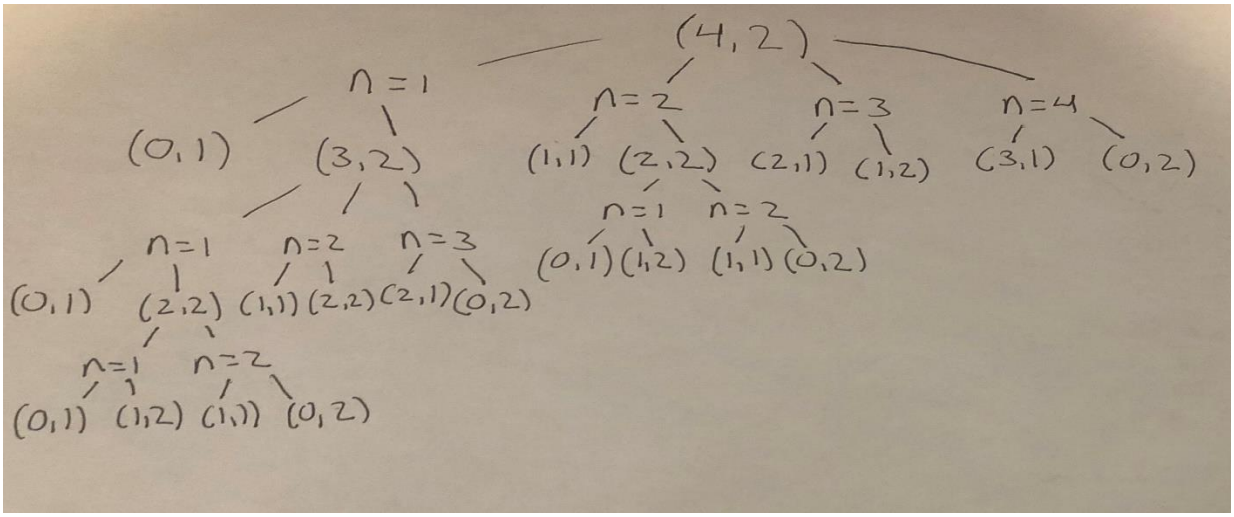
1. Falling Glass

(a) Describe the optimal substructure/recurrence that would lead to a recursive solution
The optimal substructure of the Falling glass problem has two cases. When we drop the glass from a floor x. the glass can break or not break. If the glass breaks we have to test again with x-1 floors. Then if the glass doesn't break we test again with x+1 floors. In the objective of calculating the highest floor x where the glass will not break we look at average results from these trails to find 'critical floor'. In both cases laid out the problem we are looking to solve is broken into smaller sub problems by a recursive solution.

(b) Draw recurrence tree for given (floors = 4, sheets = 2)




(c) Github

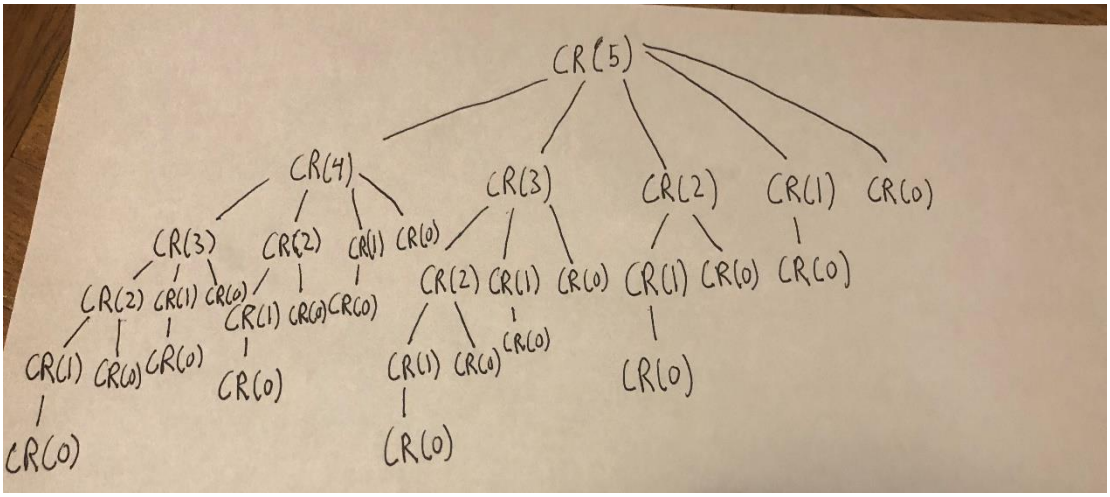
(d) How many distinct subproblems do you end up with given 4 floors and 2 sheets?
End up with 8 distinct sub problems.
(0,1), (1,1),(1,2), (2,2), (3,2), (2,1), (3,1), (0,2)

(e) How many distinct subproblems for n floors and m sheets?
The falling glass problems creates m * n distinct subproblems.

(f) Describe how you would memoize GlassFallingRecur
You memoize glassfallingrecur by creating an array 2D array ex: Memarray[i][j] and store the created subproblems in the array. Then during recursion if there's an answer already solved within the array the recursion wouldn't have to recalculate that particular subproblem.

2. Rod cutting

A. Draw the recursion tree for a rod of length 5




B. On page 370: answer 15.1-2 by coming up with a counterexample, meaning come up with a situation / some input that shows we can only try all the options via dynamic programming instead of using a greedy choice.

Let's say:

Rod length n = 4

p1= 1 p2=6 p3=7 p4=2

Using the Greedy Algorithm it will first choose p3 because it has the largest value of 7, then it will take the remaining rod p1 which is 1 for a total value of 8. This isn't the optimal solution because it's not the largest value possible. The optimal solution is to cut the rod into two length of p2 which would in turn result in a value of 12.