Dynamic Programming Assignment

1. Falling glass

a) Optimal substructure / recurrence

Say we drop a glass from floor x, then we can only have two possible cases: (we start off with k # of floors, n # of eggs)

i) if glass breaks

then

we don't need to check floors upper than x, if there are glass sheets left we can use.

So, problem reduce to x−1 floors, n−1 glass sheets.

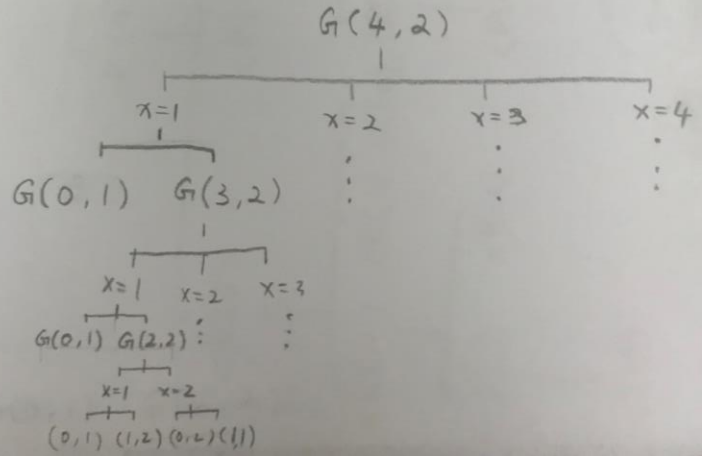
ii) if glass does not break

then

we only need to check for the floors higher than x. if there are any glass sheets left.

So, problem reduces to k − x floors, since glass sheet doesn't break, the # of glass sheets remains the same: n glass sheets.

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



d) How many distinct subproblems do you end up with given 4 floors and 2 sheets?
There are 8 distinct subproblems.

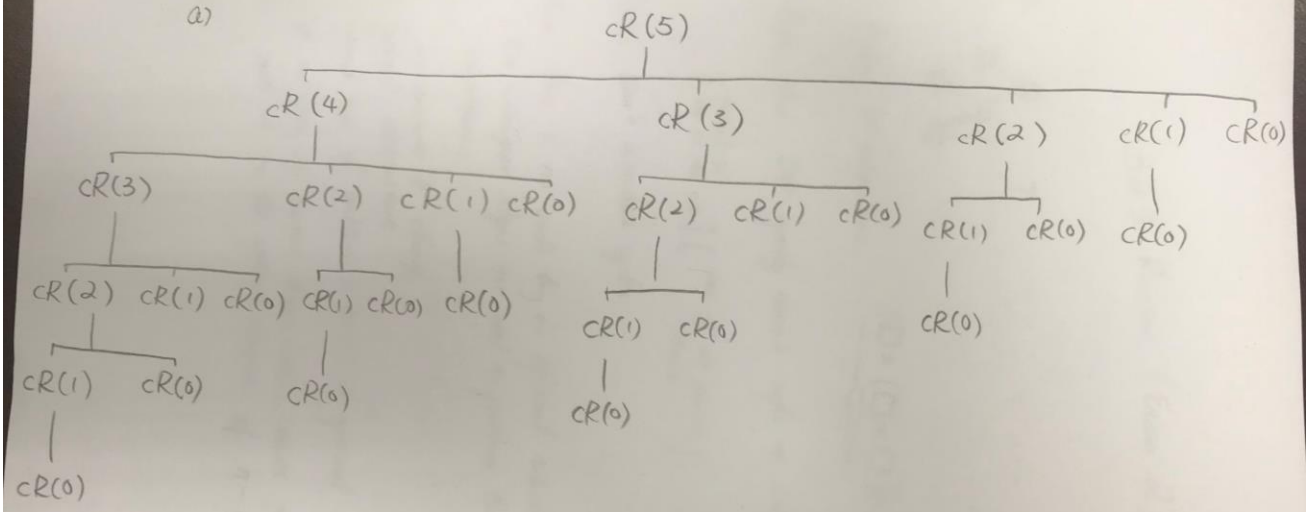
e) $n \times m = nm$

f) Describe the memoized function.

1. Check if answer already exist in memo, if so return the answer.
2. There are 3 conditions to check:
   ① if floor testing is 0^{th}
   ② if only 1 flow to be tested
   ③ if there is only one glass sheet we have to test
   return num of floors
3. Start make up a simulation situation and test the glass falling in this simulator.

2. Rod Cutting

a)



b) greedy approach: let $P_1 = 1$, $P_2 = 5$, $P_3 = 8$, and $P_4 = 9$; $n = 4$

$8/3 \approx 2.667$ has max density in greedy approach.

this approach first cut off a piece of length of 3, then the remaining length would be 1.

$$8 + 1 = 9$$

However, the better solution would be cut the rod with length 2 and it gives us $5 + 5 = 10 > 9$.