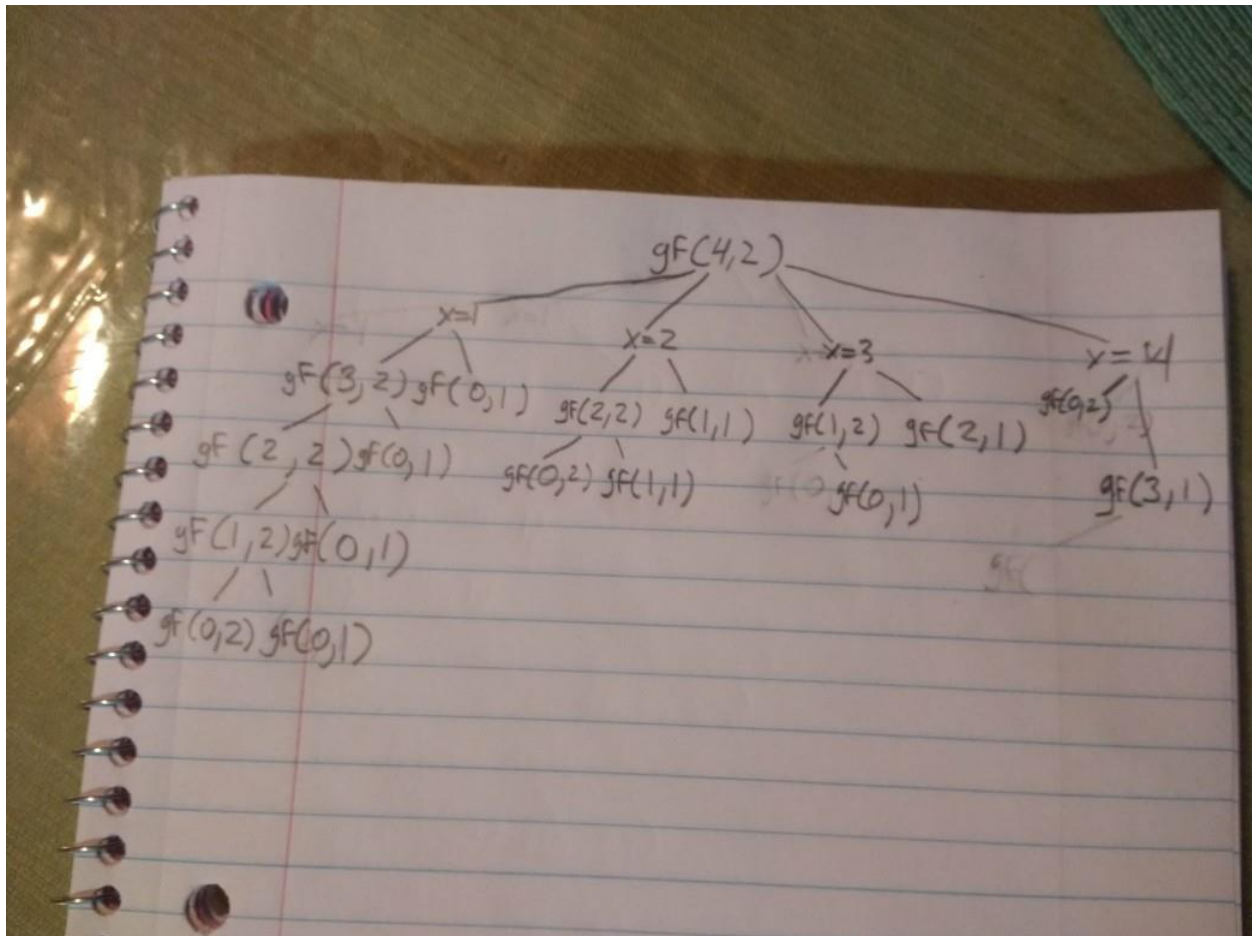


Sammy Baez

1.

a. Given n floors and m sheets, to find the minimum amount of trials you will check each floor and consider two cases, the sheet breaks or doesn't break. Let C be the current floor, if the sheet breaks then you have to check the floors beneath the current floor so the subproblem becomes $(C-1, m-1)$. If the sheet doesn't break then we continue moving up in floors until the critical floor is found, so the subproblem becomes $(n-C, m)$. To get the minimum amount of trials we will compare the maximum of these two cases and add $+1$ to signify a trial was done. Whichever case yields the lowest amount of trials will give us our answer.

b.



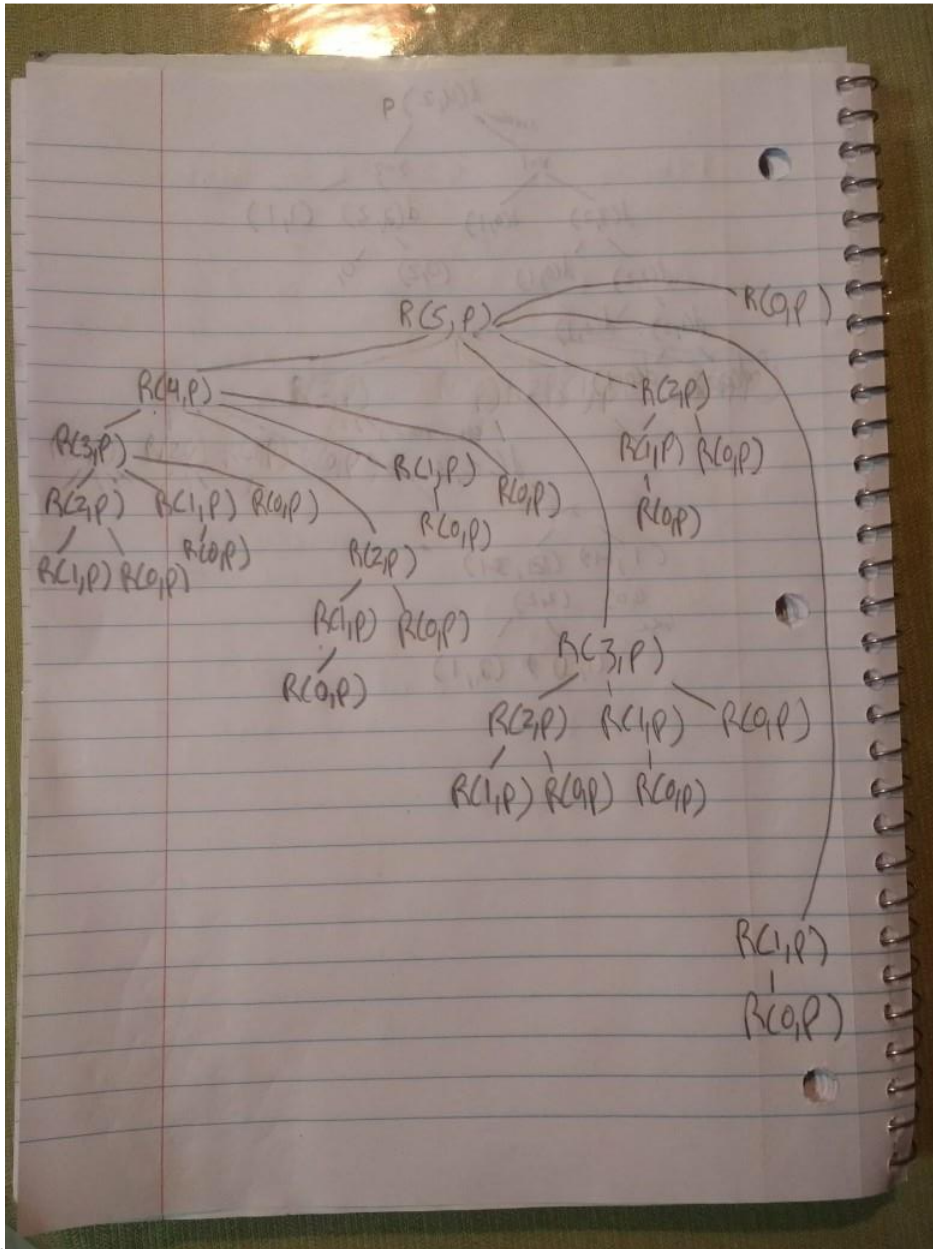
d. There are 8 distinct subproblems

e. For n floors and m sheets there are $((n-1, m-1) + (n - \text{attempt floor}, m) - \text{memo}[n][m])$ total subproblems

f. To memoize GlassFallingRecur I would initialize a 2D array where every value is the maximum an integer can hold. GlassFallingRecur would check that array and see if a solution to the current subproblem is already in the array. It would do this by checking if the array index it is currently looking

at is not the maximum integer. If it is the maximum integer then a solution to that subproblem has already been found and that solution is returned which will prevent repeat computation.

2.



a.

b. Consider an array representing a rod with length 5 containing the price amount at each length,

length: 1 2 3 4 5
array = {1, 45, 22, 15, 31}

If the greedy strategy chooses a rod of length 4, then we get a price of 15 for the rod of length 4 plus 1 for the rod of length 1 which gives us a total price of 16. But the optimal solution is to cut the rod into two pieces of length 2 and one piece of length 1 for a total price of 91. So the greedy strategy did not determine the optimal solution in this case.