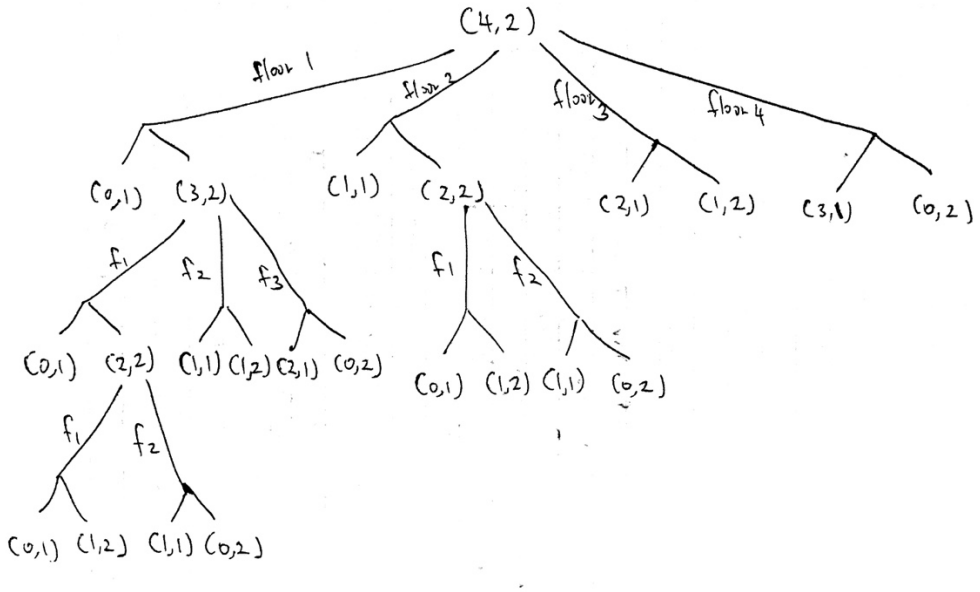
Jingwei Chen

1. Falling Glass
   (a) In the worst case, to guarantee getting the right result we have to try all the floors from lowest floor because if we try from the highest floor with one glass and it breaks we will no longer have glass to try the other floors which the result is not guaranteed, and for every floor the glass will either break or don't break, if we just have 1 egg we will just try from the lowest to highest which the result will have to include all the floors, for 0 floor there is no trial, for 1 floor there just 1 trial, assume we have n floor, m glasses, if glass break at n floor, we lost 1 glass and have to check the floor below, this sub solution will be (n - 1, m - 1) trials left to check, if the glass does not break at n floor, the amount of glass does not change, but we can check further floor, this sub solution will be (n, highest floor - m) trials left to check because highest floor - m will get me the number of floor left to reach the highest floor. For every floor of these two possibility we choose the all the highest trial because we want the guaranteed result, and then since the question asking for minimum trial we choose the minimum trial from these floors.

   (b) Recurrence tree



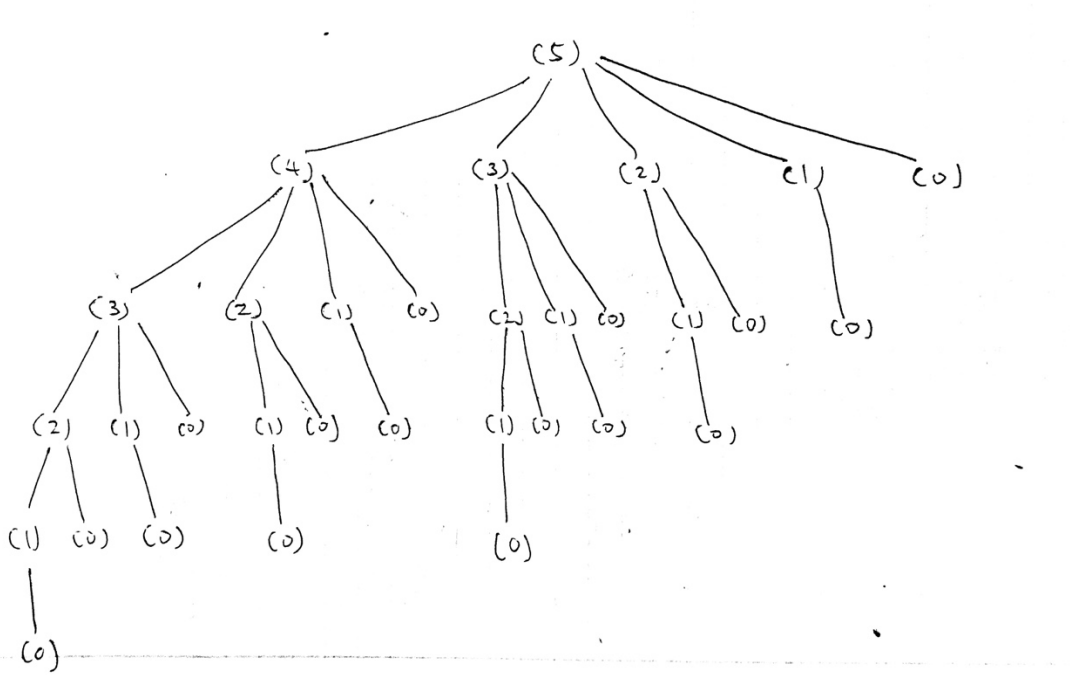
   (c) GlassFalling.java
   (d) 8 distinct subproblems
   (0,1) (3,2) (1,1) (2,2) (2,1) (1,2) (3,1) (0,2)
   (e) N*M distinct subproblems

(f) We can use 2d array or map for memoization, for using map I will convert all the floor and sheets to string, use a Map<String,Integer>, check the condition first if map.containsKey(floor+sheet) if true use the solution map.get(floor+sheet), if does not contain the key , do the calculation first then use map.put(floor+sheet, solution) to record the answer.

(g) GlassFalling.java

2. Rod Cutting

(a) Recurrence tree



(b)

length	1	2	3	4	5
price	6	30	42	64	65
density	6	15	14	16	13

Given length 5, if we use the greedy algorithm, it will cut from length 4 since it has the highest density, and left with length 1 which the total price is 64+6=70, but this not the optimal solution, length 3 and 2 gives price 30+42=72 which has greater value, therefore it is better to use dynamic programming to try all the options.

(c) RodCutting.java

(d) RodCutting.java