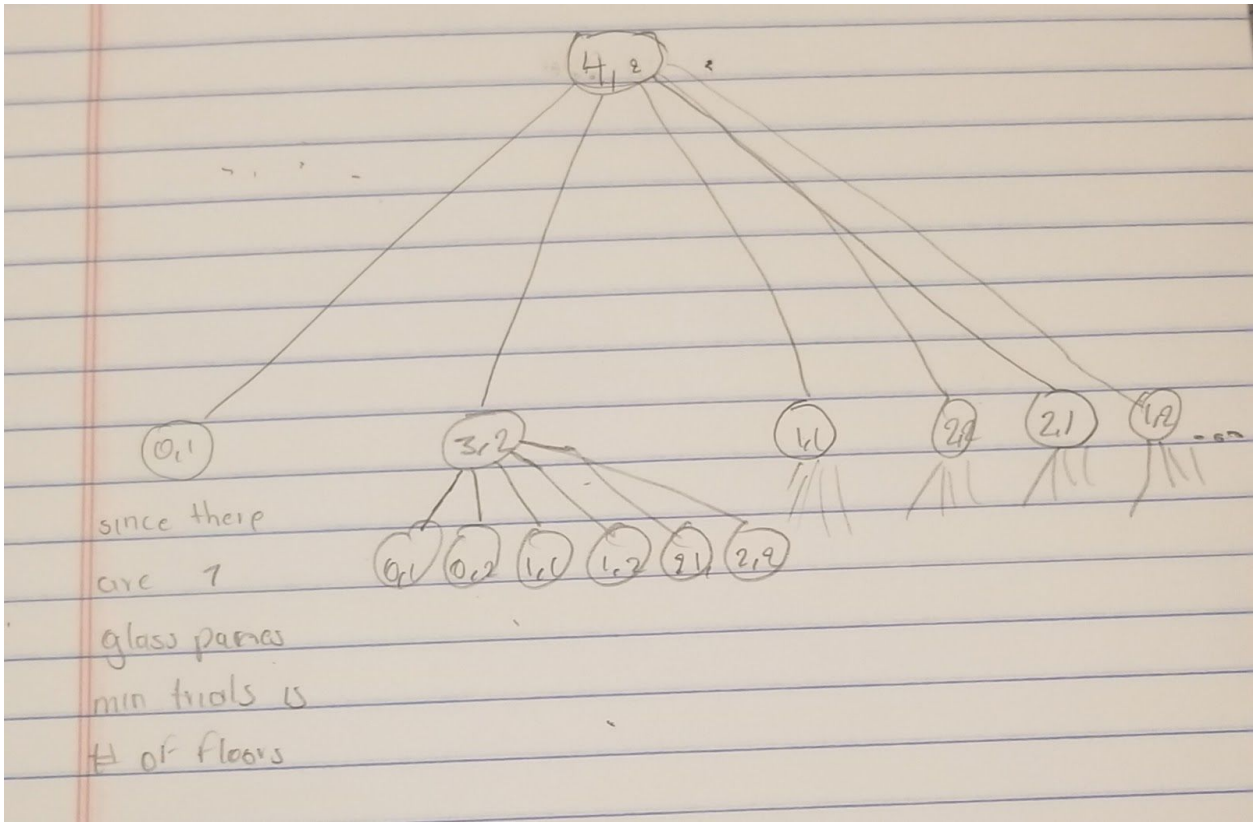
Nelson Zeas

GLASS SHEETS

a) Dynamic programming can be used to find the minimum number of trials that it would take to find if a glass shatters when it falls from a certain floor. Let i represent the number of floors and m the number of glass panes. If there is only one floor, then the number of trials is 1 because we only need to test from one floor or 0 if there are no floors. If there is only one glass pane, the the number of trials would be equal to the number of floors, since we would have to test drop from each floor starting from floor 1 to ith. When a glass pane is being dropped from the ith floor, the glass pane either shatters or survives the fall. If the glass shatters, then we have to test the lower i-1 floors. If the glass shatters, then we have to test the above floors with a new pane of glass.

b)



c) It is coded in file GlassFalling.java

d) There are 8 distinct subproblems
(0,1), (0,2), (1,1), (1,2), (2,1), (2,1), (3,1), (3,2)

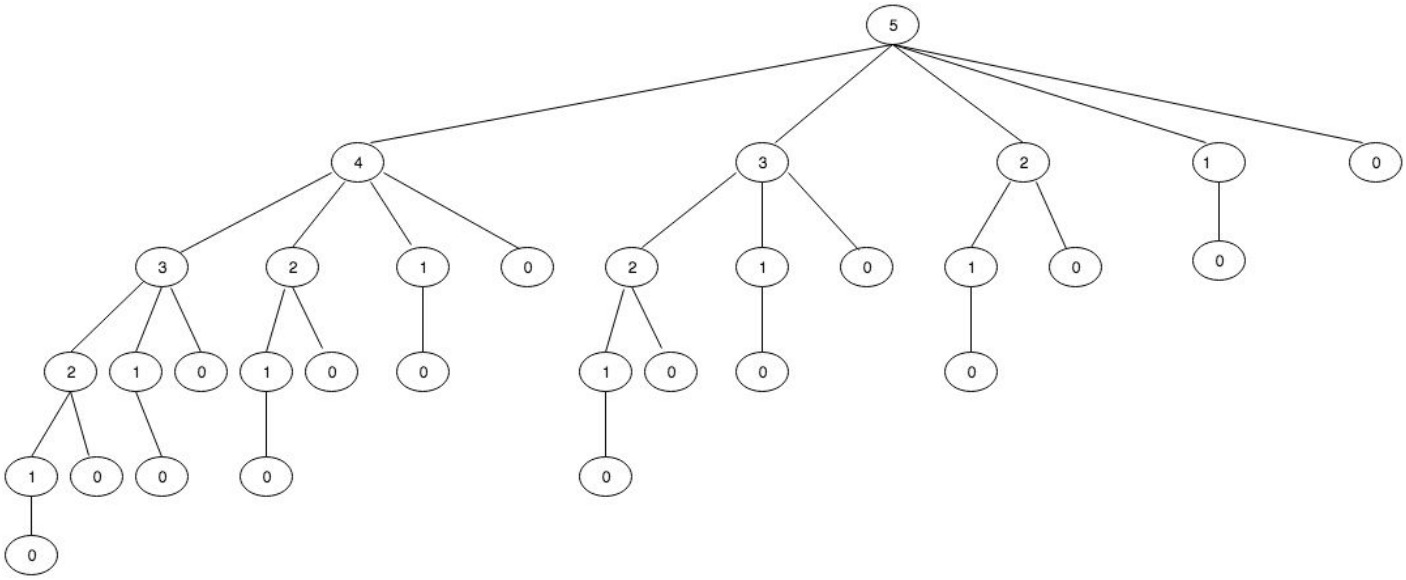
e) There are (n*m) distinct subproblems

f) A 2-D array can be used. The size of the 2-D array table should be table[sheets][floors]. The method would follow the same instructions as the recursive method except, before calling itself again, it would check the table if a value exists. If a value exists, it does not need to call itself again, but if the value is null, then it needs to call itself.

g) It is coded in file GlassFalling.java

ROD CUTTING

Recursion tree



Counter example

Length R	1	2	3	4
Price Pi	1	20	33	36
Pi/i	1	10	11	9

With a rod of length 4, we first cut out a rod of length 3 for a price of 33, which leaves us with a rod of length 1 of price 1 as described by the greedy strategy in problem 1-2 in the book. The total price for the rod is 34. The optimal way is to cut it into two rods of length 2 each fetching us 40 dollars.