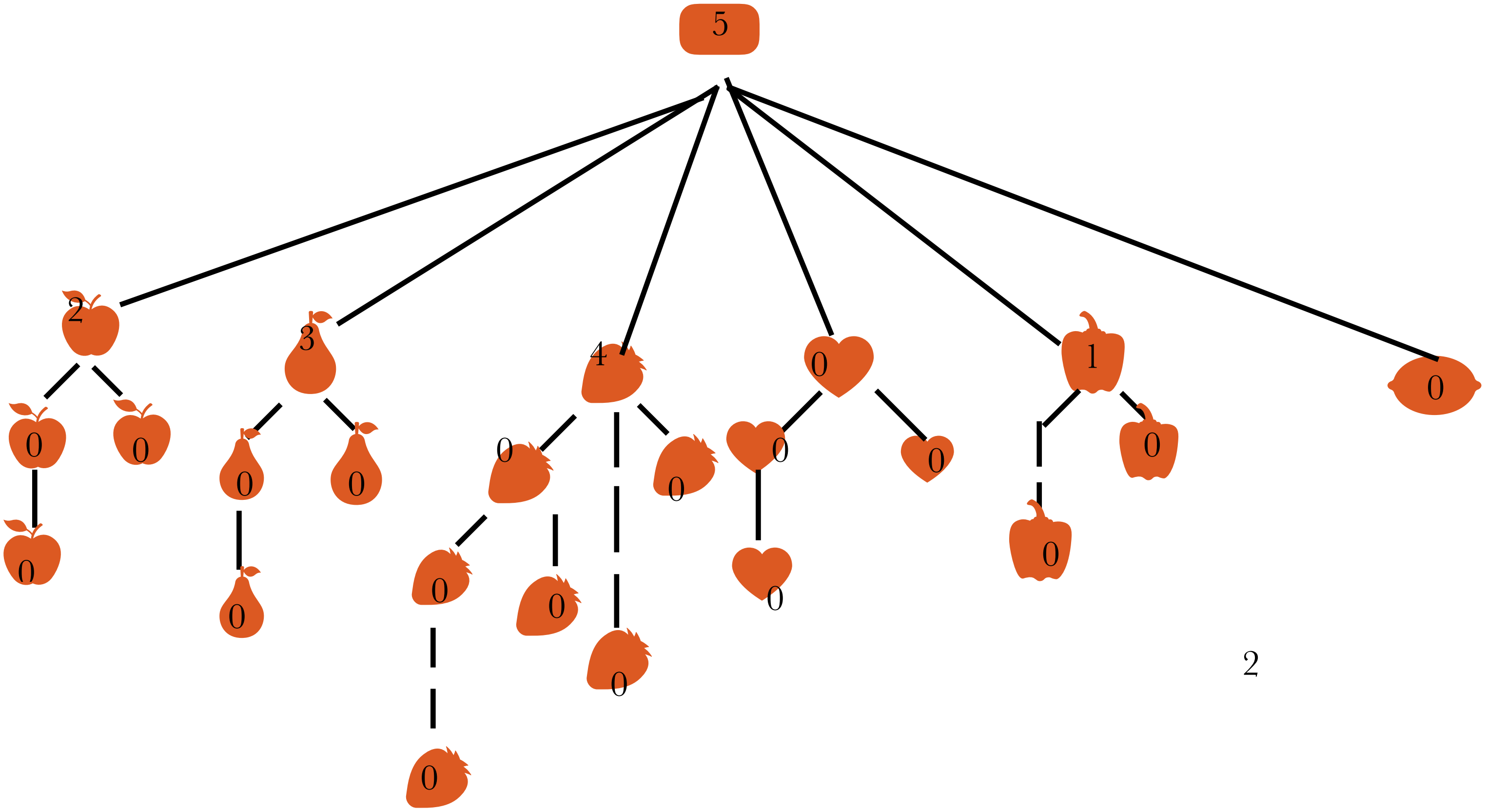
Jesse Rosenthal
Professor Yao
CSCI 323
March 30, 2019

Rod Recursion Tree

A)

Recursion Rod length 5



B) page 370

Question 15.1 -2 —— Come up with counter example, meaning come u with a situation / some input that shows we can only try all the options via dynamic programming instead.

First lets understand the process and reason for the Rod cutting. The rod cutting problem is meant to solve the maximum amount of revenue cost when given a rod of n length with m inches where all price values are contained in m inches and m does not exceed n. So if I have a rod say n = 8 with m inches with price values each starting from 1 to 8 in that order ; we have m price values as 1 2 3 4 5 6 7 8

(1, 2, 5, 9, 10, 15, 15, 18)

If I cut this imaginary rod in pieces the maximum value would be 20 when cut in two pieces 3 and 6.

Now say I work for an AI gaming company that allows users to pay for block time service to use their internet gaming platform. I am ask to create a program that efficiently generates the maximum price value per user blocked time given n hours. So if I cut this block time randomly in pieces and continuing this process recursively it is likely that I would have recalculated values. So why don't we just use Dynamic programing to store these price value results in an array such that the calculated values are stored. Now my program won't have to calculate twice for a result that has already been generated. It would simply just gather the data result already stored and reveal it therefore optimizing my program to run O(n^2)

CODES:

C) MemorizeCuttingRod.java

D) RodCuttingBottomUp.java