

# Algorithm Analysis

*Reagan Shirk*

*September 15, 2020*

## Divide and Conquer

### Stock Market Example

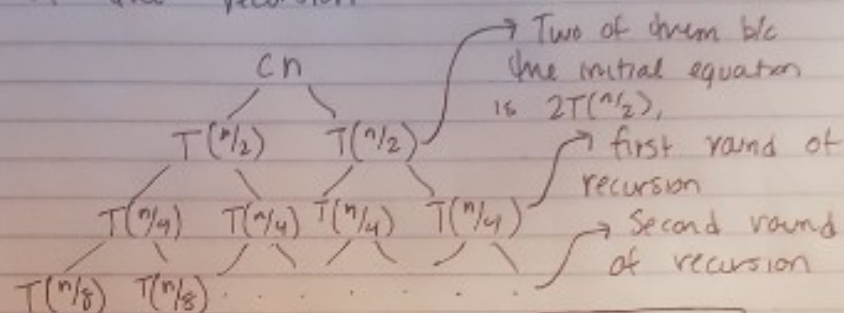
- If you divide and conquer for the stock market example, you basically split the array in half and take one of three options:
  - buy and sell in the first half (recursive)
  - buy and sell in the second half (recursive)
  - buy in the first half and sell in the second half
    - \* you want to buy at the lowest point and sell at the highest point
    - \* greedy is the best approach here (do I know what that means? Not really)
  - What is your output?
    - \* the best among the three possibilities listed above
  - Complexity:
    - \*  $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + cn$

## Recursion Trees

- On the quiz

Create a recursion tree for  $T(n) = 2T(n/2) + cn$

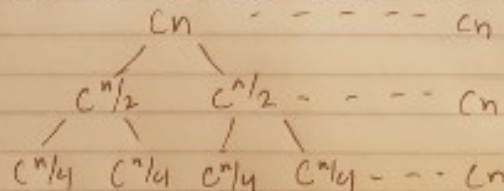
- The root of the tree should be the last term in the equation
- Each level of the tree represents one iteration of the recursion



Each layer of the tree is getting divided by 2 because the initial fraction is  $n/2$  there are two children on every node because the initial coefficient is 2.

What is the height of the tree? It will always be  $\log_x(n)$  where  $x$  = the denominator of the initial fraction. In this case,  $\log_2(n)$  or  $\lg(n)$

What is the width of the tree? Turn every  $T()$  on the tree to  $c$  and count the number of  $c$



The width of the tree will be  $cn$ , because the bottom most level will sum to be  $cn$ . The complexity is the width  $\cdot$  height, so:

$$T(n) = cn \lg(n) = \Theta(n \lg(n))$$

## Strassen's Algorithm

- Very important, clever algorithm for matrix multiplication
- Apparently we really need to read the book before lecture, I have yet to do that.
- Cheng just asked if we all knew what a matrix was. You know, just in case some senior computer science students hadn't ever heard of one before.
- Probably going to upload pictures of this too, I do know how to write matrices in LaTeX but it takes forever
- Time complexity of doing matrix multiplication using the standard dot product method
  - Number of multiplications:  $n^3$
  - Number of additions:  $n^3 - 1$
  - Number of entries in the square matrix:  $n \times n$
  - Total complexity:  $\Theta(n) \times n^2 = \Theta(n^3)$ 
    - \* pretty shitty if you ask me
- Can we do better than this? I really fucking hope so
- Strassen's algorithm is better than this and is good to use in practice but still isn't the best
- Strassen's is divide and conquer, you have an  $n \times n$  matrix that you divide into four even quadrants
  - You can't improve the matrix addition part, that doesn't get any better than  $\Theta(n^2)$ , so you want to improve the matrix multiplication