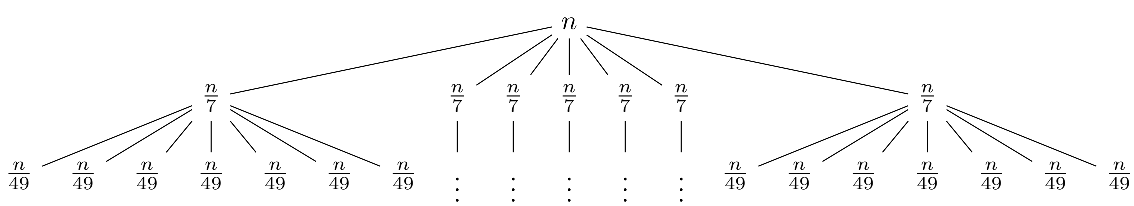
CSCI 305 Assignment 3

(Solo) Isaac Boaz

November 2, 2023

1. Provide a Θ bound for the solution of each of these recurrences.

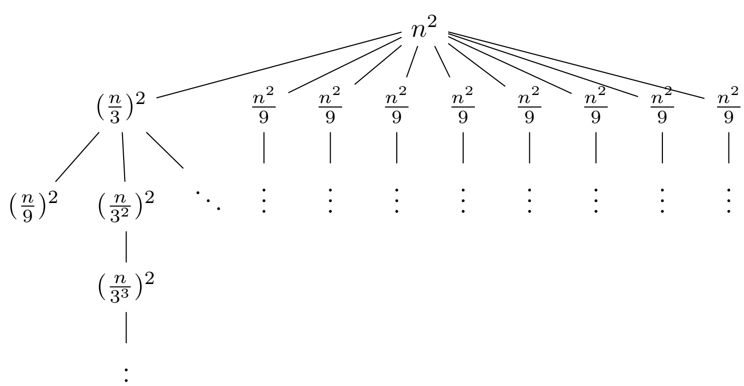
    1. $$T(n) = 7T(n/7) + n$$

        We see each level has $7^i \cdot \frac{n}{7^i} = n$ work done. Since n is being divided by 7 each level, this will be run $\log_7 n$ times.

        $$n \cdot \log_7 n \to \Theta(n \log n)$$

    2. $$T(n) = 9T(n/3) + n^2$$

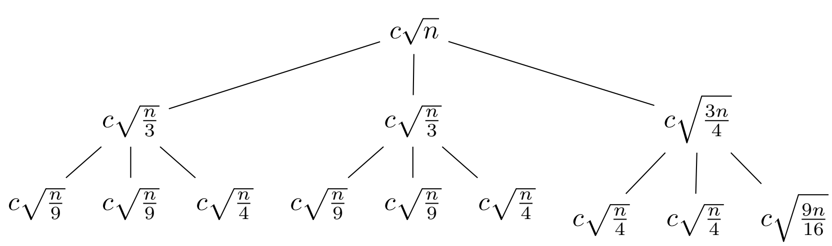
        

        At each level we do n^2 amount of work. Since we're dividing by 3 each time, we will do $\log_3 n$ levels. Thus, our runtime is $n^2 \cdot \log_3 n \to \Theta(n^2 \log n)$

    3. $$T(n) = 49T(n/25) + n^{3/2} \log n$$

2. Draw the recurrence tree for the following recurrence:

    $$T(n) = 2T(n/3) + T(3n/4) + c\sqrt{n}$$

3. FFT

   1. Give an asymptotic Θ-bound for lines 1-3.

   $$\Theta(1)$$

   2. Give an asymptotic Θ-bound for lines 6-9.

   $$\Theta(n)$$

   3. What size array is being input and output?
   We see the first array takes the Fourier Transform of the even-indexed elements, and the second array takes the Transform of the odd-indexed elements. Thus, each array is $n/2$ in size.

   4. Recurrence relation for the cost of $T(n)$.

   $$\begin{aligned} T(n) &= 2T(\frac{n}{2}) + O(\frac{n}{2}) \\ &= 2T(\frac{n}{2}) + c\frac{n}{2} \end{aligned}$$

   5. Solve the above recurrence relation.

   $$cn/2$$
   $$cn/4 \qquad cn/4$$
   $$cn/8 \quad cn/8 \quad cn/8 \quad cn/8$$

   Going by the tree diagram, we see each level x has $\frac{n}{2}$ work. Since we halve the size of the problem each level, there will be $\log_2 n$ levels, amounting to a runtime of

   $$O(n \log n)$$

   Additionally, since the subtree is balanced, we can say it is both $O(n \log n)$ and $\Omega(n \log n)$, and thus $T(n) = \Theta(n \log n)$.

   6. Find the Θ cost of slowFT. Since the outer for loop is run n times, and the inner loop is consequently run n^2 times, we know that *Algorithm 1* is asymptotically faster.

   $$\Theta(n^2) > \Theta(n \log n)$$

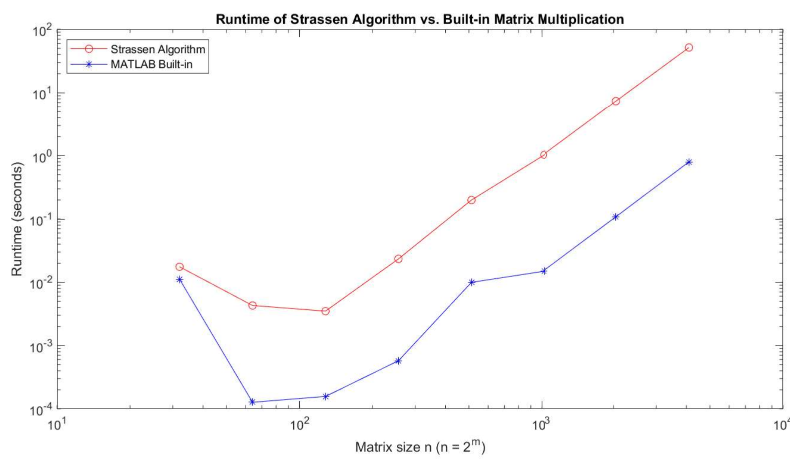
4. Strass

   1. The base case for this algorithm is when it encounters matrices that are 16×16 or smaller.

   2. I have verified this.

   3.

n	$s(n)$	$m(n)$
32	0.0176	0.0112
64	0.0043	0.0001
128	0.0035	0.0002
256	0.0234	0.0006
512	0.1993	0.0100
1024	1.0313	0.0150
2048	7.3894	0.1085
4096	51.6236	0.7990


Runtime of Strassen Algorithm vs. Built-in Matrix Multiplication
Strassen Algorithm
MATLAB Built-in
Runtime (seconds)
Matrix size n (n = 2^m)
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^1
10^2
10^3
10^4