Assignment 1

Aayush Goyal - EE18BTECH11001

Github repository

https://github.com/aayush2710/EE4013–Assignment1

1 Problem

Consider the following Recurrence Relation

$$T(n) = \begin{cases} T(n/2) + T(2n/5) + 7n & \text{if } n > 0 \\ 1 & \text{if } n = 0 \end{cases} \tag{1.0.1}$$

Which one of the following options is correct?

1) $T(n) = O\left(n^{\frac{5}{2}}\right)$
2) $T(n) = O(n \log n)$
3) $T(n) = O(n)$
4) $T(n) = O\left((\log n)^{\frac{5}{2}}\right)$

2 Solution

Let us start by drawing the recurrence tree for the given problem. We can calculate the complexity of each level by summing up the complexity of each node at that level.
For Example : At L1, complexity is
$7\left(\frac{n}{2}\right) + 7\left(\frac{2n}{5}\right) = 7\left(\frac{9n}{10}\right)$

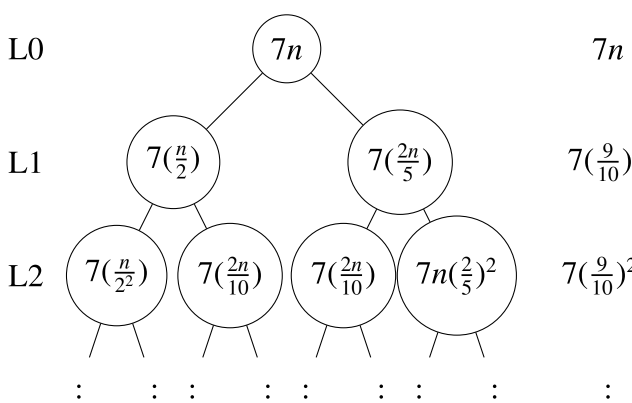

Fig. 1: Recurrence Tree

Length of left most subtree = $\log_2 n$
Length of right most subtree = $\log_{5/2} n$
Rightmost subtree is shortest so it determines the lower bound of time. Whereas the leftmost subtree is the longest, hence it determies the upper bound of time.
Height of the leftmost subtree is : $\log_2 n$ and so

$$T(n) \leqslant 7n + 7n\left(\frac{9}{10}\right) + 7n\left(\frac{9}{10}\right)^2 + \cdots + 7n\left(\frac{9}{10}\right)^{\log_2 n} \tag{2.0.1}$$

$$T(n) \leqslant 7n\left(1 + \frac{9}{10} + 7\left(\frac{9}{10}\right)^2 + \cdots + 7\left(\frac{9}{10}\right)^{\log_2 n}\right) \tag{2.0.2}$$

$$T(n) \leqslant 7n\frac{1 - \left(\frac{9}{10}\right)^{\log_2 n+1}}{1 - \frac{9}{10}} \tag{2.0.3}$$

$$T(n) \leqslant 70n - 70\left(\frac{9}{10}\right)\left(n^{\log_2\left(\frac{9}{10}\right)}\right) \tag{2.0.4}$$

$$= 70n - 63n^{0.85} \tag{2.0.5}$$

$$\implies T(n) \leqslant 70n \tag{2.0.6}$$

$$\implies T(n) \in O(n) \tag{2.0.7}$$

Height of the rightmost subtree is $log_{5/2}n$ and so,

$$T(n) \geqslant 7n + 7n\left(\frac{9}{10}\right) + 7n\left(\frac{9}{10}\right)^2 + \cdots + 7n\left(\frac{9}{10}\right)^{\log_{5/2} n} \tag{2.0.8}$$

$$T(n) \geqslant 7n\left(1 + \frac{9}{10} + 7\left(\frac{9}{10}\right)^2 + \cdots + 7\left(\frac{9}{10}\right)^{\log_{5/2} n}\right) \tag{2.0.9}$$

$$T(n) \geqslant 7n\frac{1 - \left(\frac{9}{10}\right)^{\left(\log_{5/2} n\right)+1}}{1 - \frac{9}{10}} \tag{2.0.10}$$

$$T(n) \geqslant 70n - 70\left(\frac{9}{10}\right)\left(n^{\log_{5/2}\left(\frac{9}{10}\right)}\right) \quad (2.0.11)$$

$$= 70n - 63n^{0.89} \quad (2.0.12)$$

$$T(n) \geqslant n\left(70 - \frac{63}{n^{0.11}}\right) \quad (2.0.13)$$

Since we consider complexity for large n, $\lim_{n\to\infty} \frac{63}{n^{0.11}} = 0$ So $T(n) \geqslant 70n$

3 Answer

- $T(n) \in O(n)$ (3.0.1)
- $T(n) \sim 70n$ (3.0.2)

Hence option 3 is correct

4 Verification

To verify the theoretical results, I constructed a recursive function in C with given recurrence relation and measure the execution time for different n.
The time taken was plotted using python with varying n, Here is the plot

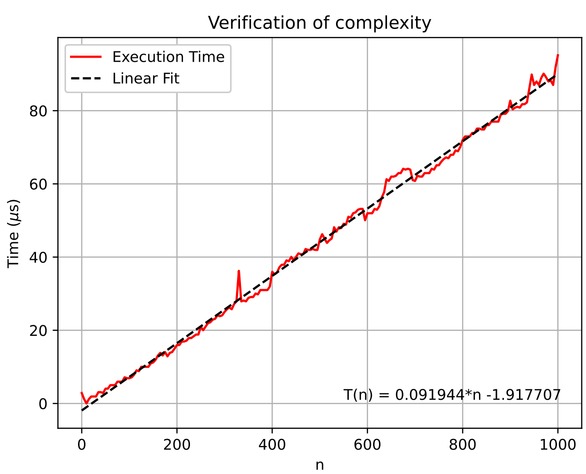

Fig. 2: T(n) vs n

We can clearly observe a linear curve. This verifies that $T(n) \in O(n)$

Using least squares, approximate time equation is
$T(n) = 0.091944n - 1.917707$

Here is the recursive function used

```
void recursive(int n) {
    if(n == 0) {
        return;
    }
    long long int Sum = 0;
    for(int i=0;i<7*n;i++){
        Sum+=i;
    }
    recursive(n/2);
    recursive(2*n/5);
}
```

This plot can be generated through the following python script

```
https://github.com/aayush2710/EE4013-
    Assignment1/code/verification.py
```