Fundamental Algorithms 4

Exercise 1

1. Try the Recursion Tree Method for the following recurrence:

   \[ T(n) = T\left(\frac{1}{3}n\right) + T\left(\frac{2}{3}n\right) + \theta(n), \]

   assuming that all occurring divisions are without remainder and \( T(1) = 0 \).

2. Show that the height of the recursion tree is in \( O(\log(n)) \).

3. What could be a flaw using the recursion tree method for such unbalanced trees?

4. Show that \( T(n) \in \theta(n \log(n)) \), anyway, by using the substitution method.

Exercise 2

For the so-called BFPRT Algorithm, an algorithm to determine the median element of an array, we obtain the following (slightly simplified) recurrence equation for its running time \( T(n) \) (depending on the number \( n \) of elements):

\[ T(n) = s(n,k) + T\left(\frac{1}{k}n\right) + T\left(\frac{1}{2k}n\right). \]

\( k \) and \( l \) are parameters (\( k \) usually small, for example \( k = 3 \) or \( k = 5 \)) where \( k = 2l + 1 \). For the function \( s \), we can assume \( s(n,k) \in \Theta(n \log k) \).

1. Show that \( T(n) \in O(n) \).

2. Does it make sense to use growing values for \( k \) (and \( l \), respectively)?