

Algorithms - Exercise 1

Due Wednesday 27/10 24:00

1. Consider the algorithm $\frac{2}{3}$ -merge-sort that is exactly the same as merge-sort with the only difference that the recursive step sorts first the first $2/3$ of the array then the last $1/3$ and then merges the two.

Write a pseudo-code for $\frac{2}{3}$ -merge-sort and analyze its running time (as a function of the length of the array).

2. Let $p_1 = \sum_{i=0}^{d_1} a_i n^i$ and $p_2 = \sum_{i=0}^{d_2} b_i n^i$ be two polynomials, where $a_{d_1} \geq 0$ and $b_{d_2} \geq 0$. prove:
 - (a) $p_1(n) = \Theta(p_2(n))$ if and only if $d_1 = d_2$.
 - (b) $p_1(n) = o(p_2(n))$ if and only if $d_1 < d_2$.
3. Rank the following functions by their order of growth. Prove your answers.

$$\log^2(n), n^{\log \log n}, n^{1/3}, n!, \log(n!)$$

4. Let $G = (V, E)$ be a tree. Prove:
 - (a) If we add an edge to G then the new graph contains a cycle.
 - (b) If we remove an edge from G then the new graph is not connected.
 - (c) There is exactly one simple path between every two vertices in G .
5. (reshut) Define the iterative logarithm function as $\log^*(n) = \min\{i : \log^{(i)}(n) \leq 1\}$, where

$$\log^{(i)}(n) = \begin{cases} n & i = 0 \\ \log(\log^{(i-1)}(n)) & \text{if } i > 0 \text{ and } \log^{(i-1)}(n) > 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

Which function grows asymptotically faster, $\log(\log^*(n))$ or $\log^*(\log n)$?