1) Determine the asymptotic runtime of the following algorithms as function of \( n \) using Big-O notation. You can assume that (...) requires constant time.

Algorithm 1.
1. \( i := 1 \)
2. While \( i \leq n \)
   3. For \( j := 1 \) To \( i \)
   4. (...) 
   5. \( i := 2i \)

Algorithm 2.
1. \( i := 1 \)
2. While \( i \leq n \)
   3. For \( j := 1 \) To \( n \)
   4. (...) 
   5. \( i := 2i \)

2) Rank the following functions by order of growth. That is, find an arrangement \( f_1, f_2, \ldots \) of the functions satisfying \( f_1 \in O(f_2), f_2 \in O(f_3), \ldots \). Partition your list into equivalence classes such that functions \( f_i \) and \( f_j \) are in the same class if and only if \( f_i \in \Theta(f_j) \).

\[
\begin{aligned}
(\sqrt{2})^{\log n} & \quad n^2 & \quad n! & \quad (\frac{3}{2})^n & \quad \log^2 n \\
2^n & \quad n^{1/\log n} & \quad \log \log n & \quad n \cdot 2^n & \quad \log n \\
2^{\log \log n} & \quad n^3 & \quad 1 & \quad 2^{\log n} & \quad (\log n)^{\log n} \\
4^{\log n} & \quad n & \quad 2^n & \quad n \log n & \quad 2^{2^{n+1}}
\end{aligned}
\]

3) Find two functions \( f(n) \) and \( g(n) \) that satisfy the following relationship. If no such \( f \) and \( g \) exist, shortly explain why.

a) \( f(n) \in o(g(n)) \) and \( f(n) \notin \Theta(g(n)) \)

b) \( f(n) \in \Theta(g(n)) \) and \( f(n) \in o(g(n)) \)

c) \( f(n) \in \Theta(g(n)) \) and \( f(n) \notin O(g(n)) \)

d) \( f(n) \in O(g(n)) \) and \( f(n) \notin O(g(n)) \)

4) Use a recursion tree to determine a good asymptotic upper bound on the recurrence \( T(n) \). Use the substitution method and (if possible) the master theorem to verify your answer.

a) \( T(n) = 3T(n/2) + n \)

b) \( T(n) = 2T(n-1) + 1 \)

c) \( T(n) = T(\alpha \cdot n) + T((1-\alpha) \cdot n) + n \)
  for some constant \( \alpha \) with \( 0 < \alpha < 1 \)
5) Joe claims he can prove that $2^n \in \mathcal{O}(1)$. His proof goes by induction on $n$.

Base case: $2^1 = 2$, i.e., $2^1 \in \mathcal{O}(1)$.
Inductive step: Assume now that $2^{n-1} \in \mathcal{O}(1)$ (Inductive Hypothesis). Then, $2^n = 2 \cdot 2^{n-1}$.
Because $2f(n) \in \mathcal{O}(f(n))$, $2^n \in \mathcal{O}(1)$.

What is wrong with Joe’s “proof”?

References