National Taiwan University
Department of Electrical Engineering
Algorithms, Fall 2019

Handout #9
October 21, 2019
TA: Chien-Hao Tsou

# Sample Solutions to Quiz #1

1. Based on Stirling's approximation, $\lg n! = \theta(n\lg n)$, $(\lg n)! = o(\lg n^{\lg n})$.
   Apply lg operation to $(\lg n)^{\lg n}$, $\lg n!$, $(\lg n)!$, $n^{1/\lg n}$, and $2^{2^n}$ respectively:
   $\lg n \cdot \lg(\lg n)$, $\lg(n \lg n)$, $\lg n \cdot \lg(\lg n)$, 1, and $2^n$
   Therefore, $n^{1/\lg n} < \lg n! < (\lg n)! \leq (\lg n)^{\lg n} < 2^{2^n}$.

2. Correct!
   $f(n) = 4n^2 - 3n + 1$
   $f(n) \leq cn^2$, for all $n \geq n_0 \Longrightarrow$ the smallest integer constant: $c = 4$.
   $0 \leq 4n^2 - 3n + 1 \leq 4n^2$, for all $n \geq n_0 \Longrightarrow$ the smallest integer constant: $n_0 = 1$ ($n_0 > 0$).
   Hence, $4n^2 - 3n + 1 = O(n^2)$ is correct.

3. (a) According to Case 3 of the Master Theorem, $a = 2, b = 4, f(n) = n^2 \lg n$.
   $f(n) = \Omega(n^{\log_4 2+\epsilon})$ for some $\epsilon > 0$, and $2(n/4)^2 \lg(n/4) \leq cn^2 \lg n$ for some $c < 1$.
   Therefore, $T(n) = \Theta(n^2 \lg n)$.

   (b) Construct the recurrence tree as Figure 1.
   $\sum_{k=0}^{\log_{3/2} n}(n) = O(n\lg n)$.
   Thus, $T(n) = O(n \lg n)$.

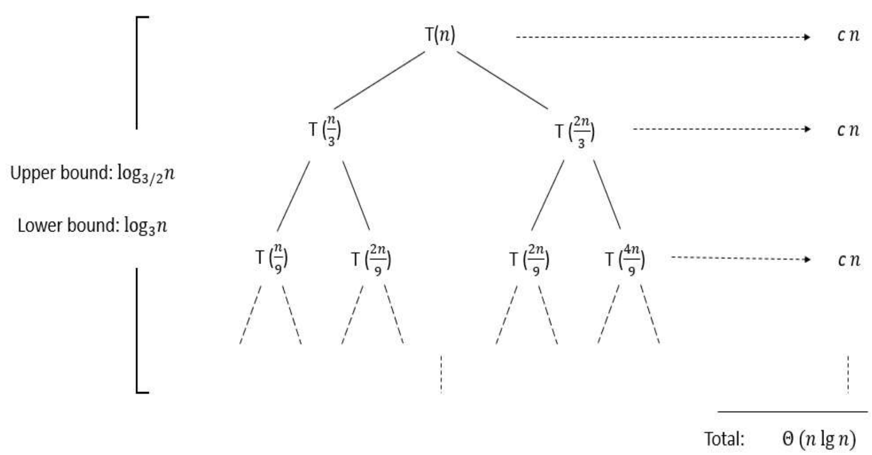


Figure 1: The recursion tree for Problem 3(b).

4. No, the algorithm does not run in polynomial time. Since the data in our computers are stored in bits, assume the proper input size of $x$ is $s$:

$$s = \lg x \Rightarrow x = 2^s.$$

The value $x$ is not a proper input size that denoting the number of bits. Thus, its time complexity is $O(x)$ and the complexity $O(x)$ can be rewritten as $O(2^s)$, which is obviously not a polynomial-time complexity.