CS 173, Fall 18 Examlet 10, Part B

Name:____

NetID:________________________ **Lecture:** **B**

Discussion: **Friday 11 12 1 2 3 4**

1. (9 points) Fill in key facts about the recursion tree for T, assuming that n is a power of 4.

   $T(4) = 7 \qquad T(n) = 2T\left(\frac{n}{4}\right) + d$

   (a) The height: $\log_4 n - 1$

   (b) Number of nodes at level k: 2^k

   (c) Sum of the work in all the leaves (please simplify):
   Each leaf contains the value 7, and there are $2^{\log_4 n - 1} = \frac{1}{2}2^{\log 4} = \frac{1}{2}\sqrt{n}$ leaves. So the sum is $\frac{7}{2}\sqrt{n}$.

   Change of base formula: $\log_b n = (\log_a n)(\log_b a)$

2. (6 points) Write the following functions in the boxes so that f is to the left of g if and only if $f(n) \ll g(n)$.

   $2^n + 3^n \qquad n^3 \qquad 100 \log n \qquad 3^{31} \qquad 3n \log(n^3) \qquad 7n! + 2 \qquad 173n - 173$

   Solution:

   $3^{31} \ll 100 \log n \ll 173n - 173 \ll 3n \log(n^3) \ll n^3 \ll 2^n + 3^n \ll 7n! + 2$

Name:____

NetID:______________________ **Lecture:** **B**

Discussion: **Friday** **11** **12** **1** **2** **3** **4**

1. (7 points) You found the following claim on a hallway whiteboard. Suppose that f and g are increasing functions from the reals to the reals, for which all output values are > 1. If $f(x)$ is $O(g(x))$, then $\log(f(x))$ is $O(\log(g(x)))$. Is this true? Briefly justify your answer.

   Solution:

   Yes, it is true. Suppose that $f(x)$ is $O(g(x))$. Then there are positive reals c and k such that $f(x) \leq cg(x)$ for all $x \geq k$. Then $\log(f(x)) \leq \log c + \log(g(x))$ for all $x \geq k$. Since $g(x)$ is an increasing function and c isn't, there is some $K \geq k$ such that $\log c \leq \log(g(x))$. So then $\log(f(x)) \leq 2\log(g(x))$ for all $x \geq K$. So $\log(f(x))$ is $O(\log(g(x)))$.

   [You don't need this much technical detail for full credit. We can ignore the other inequality from the definition of big-O because the two functions always output positive values.]

2. (8 points) Check the (single) box that best characterizes each item.

   $T(1) = d$
   $T(n) = T(n-1) + n$

$\Theta(\log n)$ ☐	$\Theta(\sqrt{n})$ ☐	$\Theta(n)$ ☐	$\Theta(n \log n)$ ☐
$\Theta(n^2)$ ☑ √	$\Theta(n^3)$ ☐	$\Theta(2^n)$ ☐	$\Theta(3^n)$ ☐

   $T(1) = d$
   $T(n) = 3T(n/3) + c$

$\Theta(\log n)$ ☐	$\Theta(\sqrt{n})$ ☐	$\Theta(n)$ ☑ √	$\Theta(n \log n)$ ☐
$\Theta(n^2)$ ☐	$\Theta(n^3)$ ☐	$\Theta(2^n)$ ☐	$\Theta(3^n)$ ☐

   $\log_5 n$ is

$\Theta(\log_3 n)$ ☑ √	$O(\log_3 n)$ ☐	neither of these ☐

   Dividing a problem of size n into m subproblems, each of size n/k, has the best big-Θ running time when

$k < m$ ☐	$k = m$ ☐
$k > m$ ☑ √	$km = 1$ ☐