Student Name:

Solution

CS2223: D-Term 2017: Quiz 2

Question 1 (Big-O & Recurrences)

(a) [2 Points] True or False: If f(n) is O(n!), then f(n) can be f(n) = n^{10} + 100 n Log n

True

(b) [3 Points] Use the Master Theorem method to solve the following recurrence and find its tight Big-O complexity.

$$T(n) = 4\,T(n/5) + n$$

$a=4$

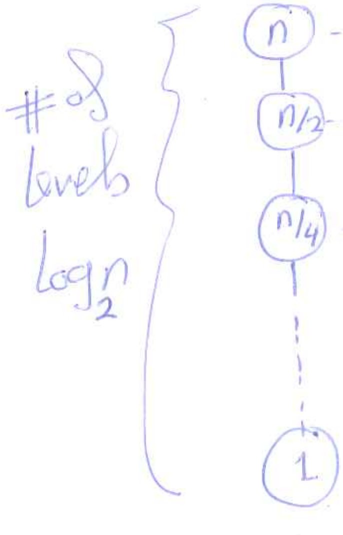
$b=5 \Longrightarrow \beta = \log_5 4 < 1 \Longrightarrow O(n^\alpha) = O(n)$

$\alpha = 1$

So $\alpha > \beta$

(c) [3 Points] Use the Tree-Based method to solve the following recurrence and find its tight Big-O complexity.

$$T(n) = T(n/2) + \text{Log}n$$



---- $\log n$

---- $\log n/2 = \log n - 1$

---- $\log n/4 = \log n - 2$

$\log n/8 = \log n - 3$

$\log n - (\log n - 1)$

Total is:

$\log n + (\log n - 1) + (\log n - 2) + \cdots (\log n - (\log n - 1))$

$\log n \Rightarrow$ repeats $\log n$ Times

So...

$\log n \cdot \log n - [1 + 2 + \cdots (\log n - 1)]$

$(\log n)^2 - [\underbrace{1 + 2 + 3 + 4 + \cdots (\log n - 1)}_{\frac{(\log n - 1)\log n}{2}}]$

$$\text{Log}^2 n - \frac{\text{Log}^2 n - \log n}{2} = \frac{\log^2 n}{2} + \frac{\log n}{2} \Longleftarrow \frac{(\text{Log}n - 1)\log n}{2}$$

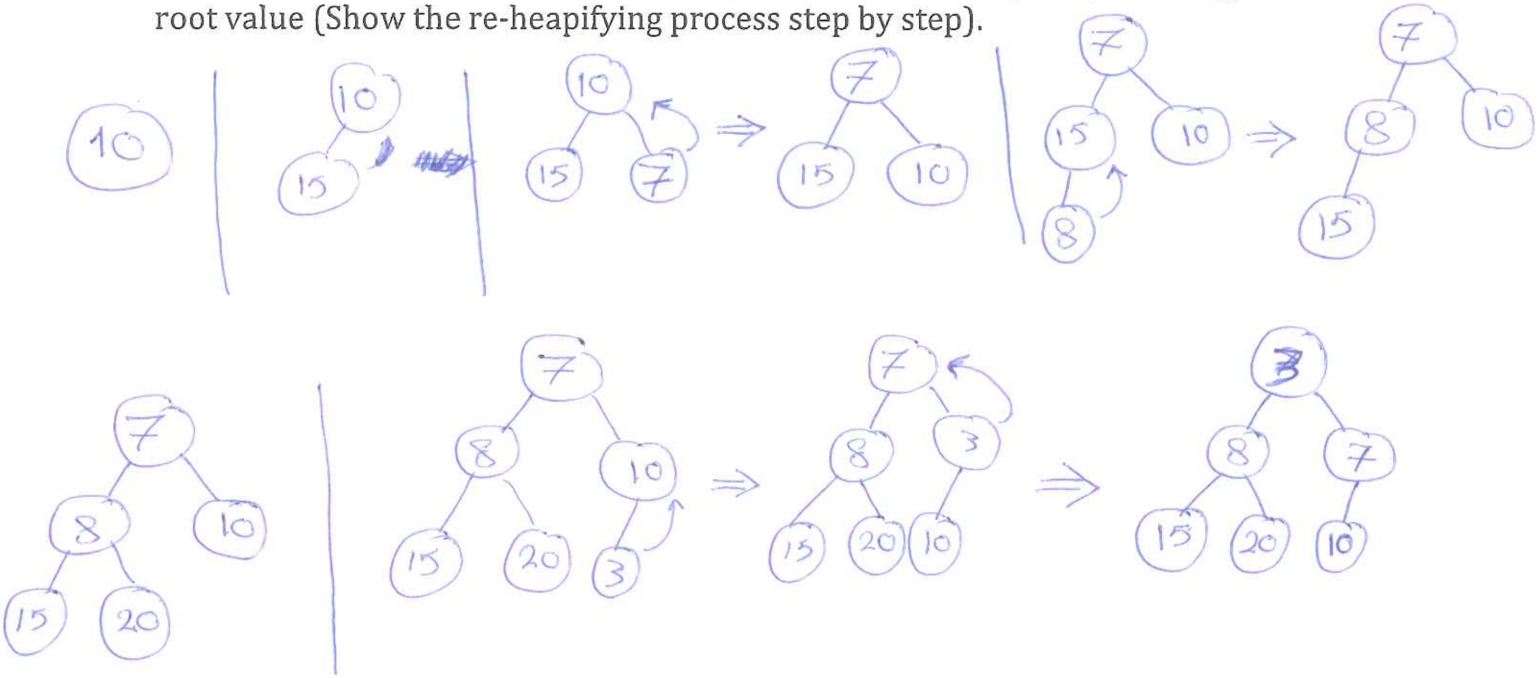
$$\Longrightarrow \boxed{O(\log^2 n)}$$

Question 2 (Heap Structure)

(a) [5 Points] Show a step by step for inserting the following values into a *min heap* structure. Show the heap after each swap in each insertion.

10, 15, 7, 8, 20, 3

(b) [5 Points] Show the array representation of the heap in the previous question. And then show the re-heapified array again after extracting out (removing) the root value (Show the re-heapifying process step by step).



3	8	7	15	20	10

Removing root

10	8	7	15	20	3

Re-heap

7	8	10	15	20	3