CSC236 Fall 2016

More Practice 05

In last lecture, we have shown that the lower bound for the worst case time complexity of the divide and conquer *binSearch* algorithm is in $\Omega(\log n)$, *i.e.*, $T(n) \geq c \log n$ where $T(n)$ is the recurrence relation for the worst case time complexity of the algorithm.

- **Example 61b.** Show that the upper bound for the worst case time complexity of the *binSearch* algorithm is in $O(\log n)$, *i.e.*, $T(n) \leq c \log n$.

  Note 1. Recall $T(n) = \begin{cases} 1 & n = 1 \\ 1 + T\left(\left\lceil \frac{n}{2} \right\rceil\right) & n > 1 \end{cases}$ worst case

  Note 2. It's important to do this practice before the lecture on Mon (Oct 17)

- **Example 61c.** Show that $e - m = \left\lfloor \frac{n}{2} \right\rfloor$. Refer to Example 61 in the slides for the definition of *e*, m, and n.
-
- **Example 62.** Using the approach in Example 61, find some pairs of c and n_0 to show that $T(n)$ is in $\Theta(\log n)$ where

$$T(n) = \begin{cases} 3 & n = 1 \\ 2 + T\left(\left\lceil \frac{n}{2} \right\rceil\right) & n > 1 \end{cases}$$

Note. We do not intend to publish solutions (or solutions outline) for any of the questions of the course notes, or extra practices. You are more than welcome to discuss your solutions with us.