

CS 161: Recitation 2 Problems

Selection, Big-O, and Recurrences

1. Recall the “Master Method” theorem from lecture:

Given a recurrence $T(n) = aT(\frac{n}{b}) + O(n^d)$ with $a \geq 1$, and $b > 1$, and $T(1) = \Theta(1)$, then

$$T(n) = \begin{cases} O(n^d \log n) & \text{if } a = b^d \\ O(n^d) & \text{if } a < b^d \\ O(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Describe the intuition for the three cases, and the intuitive interpretation of the corresponding bounds on $T(n)$ for those three cases.

2. Find a runtime bound for $T(n) = 3T(\frac{n}{2}) + O(n^2)$, with $T(1) = 1$.
3. Find a runtime bound for $T(n) = T(n - 1) + O(\frac{1}{n})$, with $T(1) = 1$
4. Our `select` algorithm from class divided the n -length array into $\frac{n}{5}$ lists of 5. Analyze the runtime of `select` if we instead divided the n -length array into $\frac{n}{7}$ lists of 7.