15-122: Principles of Imperative Computation

Recitation 7 Josh Zimmerman

Practice!

1. Rank these big-O sets from left to right such that every big-O is a subset of everything to the right of it. (For instance, $O(n)$ goes farther to the left than $O(n!)$ because $O(n) \subset O(n!)$.) If two sets are the same, put them on top of each other.

$O(n!)$ $O(n)$ $O(4)$ $O(n\log(n))$ $O(4n+3)$ $O(n^2+20000n+3)$ $O(1)$ $O(n^2)$ $O(2^n)$ $O(\log(n))$ $O(\log^2(n))$ $O(\log(\log(n)))$

Solution:

$O(4)$	$O(\log(\log(n)))$	$O(\log(n))$	$O(\log^2(n))$	$O(n)$	$O(n\log(n))$	$O(n^2+20000n+3)$	$O(2^n)$	$O(n!)$
$O(1)$				$O(4n+3)$		$O(n^2)$		

2. Using the formal definition of big-O, prove that $n^3 + 300n^2 \in O(n^3)$.

Solution: $n^3 + 300n^2 \leq n^3 + 300n^3$ for all $n > 1$. $n^3 + 300n^3 = 301n^3$. So, for all $n > 1$, $n^3 + 300n^2 \leq 301n^3$. We have $n_0 = 1$, $c = 301$ if we want to plug back in to the formal definition.

3. Using the formal definition of big-O, prove that if $f(n) \in O(g(n))$, then $k * f(n) \in O(g(n))$ for $k > 0$.

One interesting consequence of this is that $O(\log_i(n)) = O(\log_j(n))$ for all i and j (as long as they're both greater than 1), because of the change of base formula:

$$\log_i(n) = \frac{\log_j(n)}{\log(i)}$$

But $\frac{1}{\log(i)}$ is just a constant! So, it doesn't matter what base we use for logarithms in big-O notation.

Solution: Since $f(n) \in O(g(n))$, we know that there exist some $n_0 \in \mathbb{R}$ and $c \in \mathbb{R}^+$ such that $f(n) \leq c * g(n)$ for all $n > n_0$.

We can multiply both sides by k to obtain $k * f(n) \leq k * c * g(n)$ for all $n > n_0$.

So, if we set $c_1 = k * c$, then we know that $k * f(n) \leq c_1 * g(n)$ for all $n > n_0$. Thus, $k * f(n) \in O(g(n))$.