CSCB63 Tutorial 1 — Asymptotic Bounds

1 common time complexity asymptotic upper bounds

Consider the following 2D table with column headers and row headers both being: $\ln(n)$, $\lg(n)$, $\lg(n^2)$, $(\lg n)^2$, n, $n\lg(n)$, 2^n, 2^{3n}. Recall that lg is log base 2 and ln is natural logarithm (base e).

In each cell (row, col) fill in `Y` if *row*'s function is Big-Oh of *col*'s function.

	$\ln(n)$	$\lg(n)$	$\lg(n^2)$	$(\lg(n))^2$	n	$n\lg(n)$	2^n	2^{3n}
$\ln(n)$	Y	Y	Y	Y	Y	Y	Y	Y
$\lg(n)$	Y	Y	Y	Y	Y	Y	Y	Y
$\lg(n^2)$	Y	Y	Y	Y	Y	Y	Y	Y
$(\lg(n))^2$				Y	Y	Y	Y	Y
n					Y	Y	Y	Y
$n\lg(n)$						Y	Y	Y
2^n							Y	Y
2^{3n}								Y

As an exercise, prove transitivity of Big-Oh:

$$f \in O(g) \land g \in O(h) \Rightarrow f \in O(h)$$

Note that even though $3n \in O(n)$, we cannot "exponentiate both sides" to infer $2^{3n} \in O(2^n)$.

2 formal proofs of asymptotic upper bounds

We will show proofs for two entries in our Big-Oh table. We will use the Big-Oh definition directly.

1. Prove $n \in O(n\lg n)$.

   Choose $c = 1$, $n_0 = 2$. Then for all $n \geq n_0$:

   $$\begin{aligned} & n \\ =& n * 1 \\ \leq& n * \lg(n) \\ =& c * n * \lg(n) \end{aligned}$$

   Thus, $\exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow 0 \leq n \leq c * n * \lg(n)$.

2. Prove $n\lg(n) \notin O(n)$.

   We will use proof by contradiction.

   Suppose $\exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow 0 \leq n\lg(n) \leq c * n$.

   Notice that replacing $\forall n \geq n_0$ with $\forall n \geq \max(n_0, 1)$ still gives a true statement since $\max(n_0, 1) \geq n_0$.

   So, we have $\exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq \max(n_0, 1) \Rightarrow 0 \leq n\lg(n) \leq c * n$.

The reason we made this replacement is because we want $n \geq 1$, to be able to divide by n:

For all $n \geq \max(n_0, 1)$:

$$
\begin{aligned}
& n \lg(n) \leq c * n && \text{divide both sides by } n \\
\iff & \lg(n) \leq c \\
\iff & n \leq 2^c
\end{aligned}
$$

So, we have $\exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq \max(n_0, 1) \Rightarrow n \leq 2^c$.

But there is clearly a counterexample: choosing $n = \max(n_0, 2^c + 1) > 2^c$ makes the statement above false. We get a contradiction.

3 common time complexity asymptotic tight bounds

Now develop a similar table for Big-Theta: in each cell (row, col) fill in `Y` if row's function is Big-Theta of col's function.

	$\ln(n)$	$\lg(n)$	$\lg(n^2)$	$(\lg(n))^2$	n	$n \lg(n)$	2^n	2^{3n}
$\ln(n)$	Y	Y	Y					
$\lg(n)$	Y	Y	Y					
$\lg(n^2)$	Y	Y	Y					
$(\lg(n))^2$				Y				
n					Y			
$n \lg(n)$						Y		
2^n							Y	
2^{3n}								Y

4 optional exercises

1. Prove $6n^5 + n^2 - n^3 \in \Theta(n^5)$ using the definition of Big-Theta.
2. Prove $3n^2 - 4n \in \Omega(n^2)$ using the definition of Big-Omega.