MATH 510 Lecture 2

Chris Camano: ccamano@sfsu.edu

August 25, 2022

Big O notation II

As a refresher K is the point in which the function in question dominates another function for $n > K$.
Generally when solving problems of this nature it is a good place to start by identifying the constant C on the other side of the inequality
This can be done by selecting the next integer after the coefficient on the term on the left hand side with the highest degree. C can be decremented to a non integer value as long as that value is greater than the largest coefficient.

$$f(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x^1 + a_0$$

so long as $C > a_k$ the function will dominate for some K.
f(x) is O(g(x)) $\iff \exists C, K \in \mathbb{R}^+$:

$$|f(n)| \leq C|g(n)| \quad \forall n \geq K$$

Big O notation functions purely as an upper bound for lower bounds we consider Big Omega notation and if a function is bounded by some other function above and below we then turn our attention to big Theta.

Definition 0.1. Change of base formula:

$$log_a(n) = \frac{log_x(n)}{log_x(a)}$$

Proposition 0.2. *Modifying the definition of Big O notation:*

$$f(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x^1 + a_0$$

so long as $C > a_k$ the function will dominate for some K.
f(x) is O(g(x)) $\iff \exists C, K \in \mathbb{R}^+$:

$$|f(n)| \leq C|g(n)| \quad \forall n \geq K$$

$$\lim_{n\to\infty} \frac{|f(n)|}{|g(n)|} \leq C \quad \forall n \geq K$$

Example:

$$\lim_{n\to\infty} \frac{|n^{100}|}{|e^n|} \leq C \quad \forall n \geq K$$

$$\lim_{n\to\infty} \frac{|100n^{99}|}{|e^n|} \leq C \quad \forall n \geq K$$

$$\lim_{n\to\infty} \frac{100!}{|e^n|} \leq C \quad \forall n \geq K$$

$$0 \leq C \quad \forall n \geq K$$

Given a polynomial function. First eliminate negative terms by expressing an inequality in which the negative terms have been removed.

$$f(x) \leq f(x)^+, \quad f(x) \in \mathbb{P}$$

Next, adjust all of the polynomial terms to share the highest degree and sum the terms to compute a leading coefficient. In general the sum of the coefficients is a good choice of C for a big O classification. This is true for all values greater than one due to the behavior of exponentating rational numbers.

constant<logathrithmic<polynomial<exponential
if(f(n)=O(g(n))) and h(n)>0 then
f(n)h(n)=O(g(n)h(n))

Definition 0.3. Big O properties

$$O(f(n)) + O(g(n)) = O(f(n) + g(n))$$

$$O(f(n))O(g(n)) = O(f(n)g(n))$$

$$O(f(n)) + g(n)) = O(max(f(n), g(n))$$

Proposition 0.4. *Example: Given a term such as*

$$f(n) = 20(n^2 + n^2 log_4 n)(4n + 3) + (17 log_3 n + 19)(n^3 + 2)$$

First take the fastest function thats increasing

$$f(n) = O(n^2 \log(n))O(n) + O(log(n))O(n^3)$$

$$f(n) = O(n^3 \log(n)) + O(n^3 log(n))$$
$$f(n) = O(n^3 \log(n))$$

Definition 0.5. Big Omega:
f(x) is Ω(g(x)) $\iff \exists C, K \in \mathbb{R}^+ :$

$$|f(n)| \geq C|g(n)| \quad \forall n \geq K$$

Definition 0.6. Big Theta f(x) is Θ(g(x)) $\iff \exists C, K \in \mathbb{R}^+ :$

$$C|g(n)| \leq |f(n)| \leq C|g(n)| \quad \forall n \geq K$$

Definition 0.7. Little o:
$$|f(n)| < C|g(n)| \quad \forall n \geq K$$

Definition 0.8. Little ω:
$$|f(n)| > C|g(n)| \quad \forall n \geq K$$

Pseduocode and Analyzing Runtime