Learning Objectives

By the end of this worksheet, you will:

- Prove statements using the definition of Big-O and its negation.
- Represent constant functions in Big-O expressions.
- Understand and use the definition of Omega and Theta to compare functions.

For your reference, here is the formal definition of Big-O:

$$g \in \mathcal{O}(f): \ \exists c, n_0 \in \mathbb{R}^+, \ \forall n \in \mathbb{N}, \ n \geq n_0 \Rightarrow g(n) \leq cf(n) \qquad \text{where } f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$$

1. **Constant functions**. As we discussed in class, *constant functions*, like $f(n) = 100$, will play an important role in our analysis of running time next week. For now let's get comfortable with the notation.

   (a) Let $g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. Show how to express the statement $g \in \mathcal{O}(1)$ by expanding the definition of Big-O.[1]

   there exists n_0, c in R_{+}, for all n in N, n >= n_0 ==> g <= c* 1

   (b) Prove that $100 + \dfrac{77}{n+1} \in \mathcal{O}(1)$.
   Note: this proof isn't too mathematically complex; treat this as another exercise in making sure you understand the definition of Big-O.
   Hint: one algebraic property of inequalities is that $\forall x, y \in \mathbb{R}^+, \ x \geq y \Rightarrow \dfrac{1}{x} \leq \dfrac{1}{y}$.

   let n0 =0 and c = 177
   OR
   let n0 = 76 and c = 101

[1] Remember that we often abbreviate Big-O expressions to just show the function bodies. "$\mathcal{O}(1)$" is really shorthand for "$\mathcal{O}(f)$, where f is the constant function $f(n) = 1$."

2. **Omega**. Recall that we can think of Big-O notation as describing an *upper bound* on the rate of growth of a function: saying "$g \in \mathcal{O}(f)$" is like saying "g grows at most as fast as f." Sometimes we care just as much about a *lower bound* on the rate of growth and for this, we have the symbol Ω (the Greek letter Omega), which is defined analogously to Big-O:

$$g \in \Omega(f): \ \exists c, n_0 \in \mathbb{R}^+, \ \forall n \in \mathbb{N}, \ n \geq n_0 \Rightarrow g(n) \geq cf(n) \qquad \text{where } f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$$

Using this definition, prove that for all $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$, if $g \in \mathcal{O}(f)$, then $f \in \Omega(g)$.

there exist n_0, c in R_+ for all n in N, n >= n_0 ==> g(n) <= c* f(n)
WTS: there exist n_1, c1 in R_+, for all n in N, n>= n_1 ==> f(n) >= c1* g(n)

c1 = 1/c, n_0 = n_1

3. **Theta**. Both Big-O and Omega are limited in the same way as inequalities on numbers. "$2 \leq 10^{10}$" is a true statement, but not very insightful; similarly, "$n+1 \in \mathcal{O}(n^{10})$" and "$2^n + n^2 \in \Omega(n)$" are both true, but not very precise.

Our final piece of asymptotic notation is Θ (the Greek letter Theta), which we define as:

$$g \in \Theta(f): \; g \in \mathcal{O}(f) \wedge g \in \Omega(f) \qquad \text{where } f, g: \mathbb{N} \to \mathbb{R}^{\geq 0}$$

Or equivalently,

$$g \in \Theta(f): \; \exists c_1, c_2, n_0 \in \mathbb{R}^+, \; \forall n \in \mathbb{N}, \; n \geq n_0 \Rightarrow c_1 f(n) \leq g(n) \leq c_2 f(n) \qquad \text{where } f, g: \mathbb{N} \to \mathbb{R}^{\geq 0}$$

When we write $g \in \Theta(f)$, what we mean is "g grows at most as quickly as f and g grows at least as quickly as f" —in other words, that f and g have the *same* rate of growth. In this case, we call f a **tight bound** on g, since g is essentially squeezed between constant multiples of f.

Prove that for all functions $g: \mathbb{N} \to \mathbb{R}^{\geq 0}$, and all numbers $a \in \mathbb{R}^{\geq 0}$, if $g \in \Omega(1)$, then $a + g \in \Theta(g)$.[2]

(Or in other words, for such functions g, shifting them by a constant amount does not change their "Theta" bound.)

there exists n0, c in R+, for all n in N, n >= n0 ==> g(n) >= c*1
g(n) = 0 doesn't work, so g(n) must eventually be non-zero.
there exist n1, c1, c2 in R+, for all n in N, n >= n1 ==> c1g(n) <= g(n) + a <= c2g(n)
c2 = 1 + a/c, n1 = n0, c1 = 1
g(n) <= g(n) (1)
a * c/a<= g(n) (2)
a <= a/c g(n) (2')
(1) + (2')
g(n) + a <= g(n) + a/c* g(n)
g(n) + a <= (1 + a/c) g(n)

[2] Here we use $a + g$ to denote the function g_1 defined as $g_1(n) = a + g(n)$ for all $n \in \mathbb{N}$.

4. **Negating Big-O**. So far, we have only looked at proving that a function *is* Big-O of another function. In this question, we'll investigate what it means to show that a function *isn't* Big-O of another.

   (a) Express the statement $g \notin \mathcal{O}(f)$ in predicate logic, using the expanded definition of Big-O. (As usual, simplify so that all negations are pushed as far "inside" as possible.)

   there exist c, n0 in R+, for all n in N, n >= n0 ==> g(n) <= c* f(n)

   for all c, n0 in R_, there exist n in N, n>=n0 and g(n) > c*f(n)

   (b) Prove that for all positive real numbers a and b, if $a > b$ then $n^a \notin \mathcal{O}(n^b)$.

   for all c, n0 in R+, there exist n in N, n>=n0 and n^a > c*n^b

Formal proof:
Assume a > b. Let c, n0 be any number in R+.
let n = ceiling (c^ (1/(a-b)) + n0.
so, n >= n0.
n > c^ (1/(a-b))
(n^(a -b))^(1/(a-b)) > c^(1/(a-b))
n^(a-b) > c
n^a > c* n^b
(magic)

Assume a > b. Let c, n0 be any number in R+.
(n^(a -b))^(1/(a-b)) > c^(1/(a-b))
n > c^ (1/(a-b)).
and n >= n0
adding them together, we can let n be:
n = ceiling (c^ (1/(a-b)) + n0