Math 416

Worksheet 3. Basics of algorithm analysis

I. Big-O Notation.

Definition (Big O, Big Ω, Big Θ)**.** Fix a function $f\colon \mathbb{N}\to\mathbb{N}$ (sometimes we abuse notation and allow functions $f\colon \mathbb{N}\to\mathbb{R}^{\geq 0}$ or others... Sorry!):

$$O(f(n)) := \left\{ g\colon \mathbb{N}\to\mathbb{N} \,\middle|\, \begin{array}{l} \exists C > 0, N_0 \in \mathbb{N} \\ \text{s.t. } \forall n \geq N_0,\ g(n) \leq Cf(n) \end{array} \right\}.$$

$$\Omega(f(n)) := \left\{ g\colon \mathbb{N}\to\mathbb{N} \,\middle|\, \begin{array}{l} \exists C > 0, N_0 \in \mathbb{N} \\ \text{s.t. } \forall n \geq N_0,\ Cf(n) \leq g(n) \end{array} \right\}.$$

$$\Theta(f(n)) := O(f(n)) \cap \Omega(f(n)).$$

Definition. An algorithm has *polynomial run time* if $T(n)$ is $O(n^d)$ for some d.

Problem 1. Discuss in less formal terms what it means for a function $g(n)$ to be $O(n^2)$, $\Omega(\log_2(n))$, or $\Theta(n)$. Can you come up with a function that is simultaneously all three?

Problem 2. Prove that an algorithm with polynomial running time has the following desirable property: doubling the input size results in slowing down the running time by a *constant* factor.

We are never going to prove something as precise as, 'the running time on inputs of size n is exactly $\sqrt{3}n^2 + 3n + 81$.'

> *There is no sense in being precise when you don't even know what you're talking about.*
>
> -John Von Neumann

Problem 3. What purpose do the constants C and N_0 serve in the definition of $O(f(n))$? Discuss in your group why we use this definition and don't simply require $T(n) \leq f(n)$ for all n.

Problem 4. Prove in each case that T is $O(f)$, by finding specific values of the constants C and N_0 as required by the definition.

(a) $T(n) = 16n^2 + 11n + 1$, $f(n) = n^2$.
(b) $T(n) = an^2 + bn + c$, $f(n) = n^2$. (Arbitrary constants a, b, c.)

Problem 5. Prove that $T(n)$ is $\Omega(f(n))$ if and only if $f(n)$ is $O(T(n))$.

Problem 6.

(a) Are logarithms with different bases asymptotically equivalent? That is, must $\log_a(n)$ be $\Theta(\log_b(n))$?
(b) What about exponentials with different bases? Must a^n be $\Theta(b^n)$?

Lemma (Important Lemma)**.**

(1) If f and g are functions such that $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c > 0$, then $f(n)$ is $\Theta(g(n))$.
(2) If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, then $f(n)$ is $O(g(n))$ but is not $\Omega(g(n))$.

Problem 7. Prove the Important Lemma, as follows.

(a) Explain why there is a constant d such that

$$\frac{d}{2} \leq \frac{f(n)}{g(n)} \leq \frac{3d}{2}$$

for all but finitely many n.

(b) Use part (a) to conclude that $f(n)$ is both $O(g(n))$ and $\Omega(g(n))$, completing the proof of part (1) of the lemma.
(c) Half of your argument for part (1) should give in part (2) that $f(n)$ is $O(g(n))$. Verify this.
(d) *Carefully* write down what it means for $f(n)$ to *not* be $\Omega(g(n))$. (I.e., negate the definition.)
(e) *Carefully* prove in part (2) that $f(n)$ is not $\Omega(g(n))$, by showing that no constants C and N_0 work in the definition of big-Ω.

Problem 8. Prove that for any function $g\colon \mathbb{N}\to[1,\infty)$, $g(n)$ is $\Theta(\lfloor g(n)\rfloor)$.

Problem 9. Use the Important Lemma and facts from calculus to prove the following.
(a) Let $p(n) = a_0 + a_1 n + \cdots + a_d n^d$ with $a_d > 0$. Then $p(n)$ is $\Theta(n^d)$.
(b) $\log(n)$ is $O(n^d)$ for every $d > 0$. (Including non-integer d, like $d = 1/2$.)
(c) n^d is $O(r^n)$ for every $r > 1$ and every $d > 0$.

Problem 10 (Some basic properties of big-O)**.** Prove the following.
(a) If $f(n)$ is $O(g(n))$ and $c > 0$ is a constant, then $cf(n)$ is $O(g(n))$.
(b) If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, then $f_1(n)f_2(n)$ is $O(g_1(n)g_2(n))$.
(c) If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, then $f_1(n) + f_2(n)$ is $O(\boxed{})$.
(Find the best bound you can and prove that it works.)
(d) If f is $O(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n))$.

II. Little o Notation.

Definition (Little o and Little ω)**.** Fix a function $f\colon \mathbb{N}\to\mathbb{N}$:

$$o(f(n)) := \left\{ g\colon \mathbb{N}\to\mathbb{N} \;\middle|\; \begin{array}{l} \forall C > 0,\ \exists N_0 \in \mathbb{N} \\ \text{s.t. } \forall n \geq N_0,\ g(n) < Cf(n) \end{array} \right\}.$$

$$\omega(f(n)) := \left\{ g\colon \mathbb{N}\to\mathbb{N} \;\middle|\; \begin{array}{l} \forall C > 0,\ \exists N_0 \in \mathbb{N} \\ \text{s.t. } \forall n \geq N_0,\ Cf(n) < g(n) \end{array} \right\}.$$

Problem 11. (Practice with little o)
(a) Which polynomials are in $o(n^3)$?
(b) Which exponential functions are in $o(2^n)$?
(c) Which logarithmic functions are in $o(\log_2(n))$?

Problem 12. Let $f\colon \mathbb{N}\to\mathbb{N}$ be a function.
(a) Prove that $o(f(n)) \subset O(f(n))$ and $\omega(f(n)) \subset \Omega(f(n))$.
(b) Prove that $\Omega(f(n)) \cap o(f(n)) = \emptyset$.