Computer Science 260 FactSheet 2017

Big Oh Rules (Facts):

1. Transitivity of Big O: $f(n) \in \mathbf{O}(g(n))$ and $g(n) \in \mathbf{O}(h(n)) \Rightarrow f(n) \in \mathbf{O}(h(n))$.
2. Strange But True Log Domination Rule: $(\log n)^r \in \mathbf{O}(n^s)$ for all constants r and s both not equal to 0.
3. Polynomial Rule: $p(n) \in \mathbf{O}(q(n))$, whenever $p(n)$ is a polynomial in n, of degree k, and $q(n)$ is a polynomial in n, of degree t, where k and t are constants and $k \leq t$.
4. Product Rule: $f_1(n) \in \mathbf{O}(g_1(n))$ and $f_2(n) \in \mathbf{O}(g_2(n)) \Rightarrow f_1(n) \cdot f_2(n) \in \mathbf{O}(g_1(n) \cdot g_2(n))$.
5. Removal of Constant Factors Rule: $f(n) \in \mathbf{O}(c \cdot f(n))$ for all constants $c > 0$.
6. Log Base is Irrelevant Rule: $\log_a n \in \mathbf{O}(\log_b n)$ for all constants $a, b > 0$ and $a, b \neq 1$.
7. Reciprocal Rule: $f(n) \in \mathbf{O}(g(n)) \Rightarrow \frac{1}{g(n)} \in \mathbf{O}(\frac{1}{f(n)})$.
8. Sum Rule: If $f_1(n) \in \mathbf{O}(g(n))$ and $f_2(n) \in \mathbf{O}(g(n))$ then $f_1(n) + f_2(n) \in \mathbf{O}(g(n))$.
9. Less-Than Rule: if $f(n) \leq g(n)$ for all n greater than some $n_0 > 0$ then $f(n) \in \mathbf{O}(g(n))$. Use this rule only when a) the $\leq$ relation is obviously true, and b) no other rule can by applied

When using the Rules to prove a Big Oh statement, be sure to refer to the Rule explicitly by name, and number the lines of your proof; and if a statement follows logically from other statements and a rule, refer to the statements by number and the rule by name.

Definition of Big Oh: If $\exists c > 0, n_0 > 0$ such that $f(n) \leq c * g(n) \quad \forall n \geq n_0$, then $f(n) \in \mathbf{O}(g(n))$.

Master Theorem

For a function $T(n)$ defined on positive integers, where $T(n) = aT(\frac{n}{b}) + f(n)$ and $f(n)$ is a positive-valued function, and constants a and b are such that $a \geq 1$ and $b > 1$, then:

1. **If** $f(n) \in \mathbf{O}(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, **then** $T(n) \in \Theta(n^{\log_b a})$.

2. **If** $f(n) \in \Theta(n^{\log_b a})$ **then** $T(n)$ is $\Theta(n^{\log_b a} \log n)$

3. 3. **If** $f(n) \in \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and there exists some $c, 0 < c < 1$ such that $af(\frac{n}{b}) < cf(n)$ when n is sufficiently large, **then** $T(n) \in \Theta(f(n))$.

Log facts

1. $\lg n \leq n \; \forall n \geq 1$, and $\log n \leq n \; \forall n \geq 1$.
2. $\log_a^d n$ is, by definition, $(\log_a n)^d$
3. $2^{\lg n} = n$

4. $\log_a n^b = b \log_a n$
5. $\log_b a = \frac{\log_c a}{\log_c b}$
6. $\log_b a = \frac{1}{\log_a b}$
7. $a^{\log_b c} = c^{\log_b a}$
8. $\log_c(a * b) = \log_c a + \log_c b$
9. $\log_4 5 = 1.161$
10. $\log_5 4 = 0.861$
11. $\log_4 3 = 0.792$
12. $\log_3 4 = 1.262$