CS3230 Cheatsheet (Midterm)
by Yiyang, AY21/22

Asymptotic Analysis

Asymptotic Notations

Big-O: $f(n) = O(g(n))$ if there exists $c > 0$ and $n_0 > 0$, s.t., for all $n \geq n_0$

$$0 \leq f(n) \leq cg(n)$$

Similar definition for **Big-Omega**, $\Omega()$.
Small-o: $f(n) = o(g(n))$ if there exists $c > 0$ and $n_0 > 0$, s.t., for all $n \geq n_0$x

$$0 \leq f(n) < cg(n)$$

Similar definition for **Small-omega**, $\omega()$.
Lastly, $f(n) = \Theta(g(n))$ iff. $f(n) = O(g(n)) \wedge f(n) = \Omega(g(n))$.

Solve Recurrence Relations

Master Theorem

For recurrence in the form of

$$T(n) = aT(n/b) + f(n)$$

there are three cases to be considered

- If $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$
- If $f(n) = \Theta(n^{\log_b a} \lg^k n)$ for some $k > 0$, then $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$
- If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some $\epsilon > 0$ and it satisfies the **regularity condition** that $af(n/b) \leq cf(n)$ for some $0 < c < 1$, then $T(n) = \Theta(f(n))$

Notes: The three cases mean whether $f(n)$ grows **polynomially** slower, around the same rate, or faster than $n^{\log_b a}$.
Notes: The regularity condition ensures the sum of sub-problems is less than $f(n)$.

Stirling's Approximation

$n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$, or asymptotically, $\lg(n!) = \theta(n \lg n)$.

Harmonic Sequence

$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + O(1)$

Other Important Asymptotic Statements

$\lg n = O(n^\alpha)$ for any $\alpha > 0$.
$x^\alpha = O(e^x)$ for any $\alpha > 0$.

Common Recurrence Relations

(AY18/19Sem2 Midterm Qn2a) $T(n) = 2T(n/2) + n \lg \lg n = \Theta(n \lg n \lg \lg n)$
(AY19/20Sem2 Midterm Qn2d) $T(n) = 4T(n/4) + n \lg \lg \lg n = \Theta(n \lg n \lg \lg \lg n)$
(AY20/21Sem2 Midterm Qn2b) $T(n) = 27T(n/3) + n^3/\lg^2 n = \Theta(n^3)$

Hashing & Fingerprint