CS3230 midterm reference

Isaac Lai

March 9, 2024

1 Asymptotic facts

$$
\begin{aligned}
e^x &\geq 1 + x \\
a^{\log_b c} &= c^{\log_b a} \\
n! &= \sqrt{2\pi n}\left(\frac{n}{3}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right) \text{ (Stirling's approximation)} \\
\log(n!) &= \Theta(n \log n) \\
\sum_{k=0}^{n} ar^k &= \frac{a(r^{n+1} - 1)}{r - 1} \text{ (Geometric series)} \\
\sum_{k=1}^{n} \frac{1}{k} &= \ln n + O(1) \text{ (Harmonic series)} \\
\lim_{x \to \infty} \frac{f(x)}{g(x)} &= \lim_{x \to \infty} \frac{f'(x)}{g'(x)} \text{ (L'Hopital's Rule)}
\end{aligned}
$$

2 Asymptotic analysis

- $f(n) = O(g(n))$ if $\exists c > 0, n_0 > 0$ such that $\forall n \geq n_0, 0 \leq f(n) \leq cg(n)$
- $f(n) = \Omega(g(n))$ if $\exists c > 0, n_0 > 0$ such that $\forall n \geq n_0, 0 \leq cg(n) \leq f(n)$
- $f(n) = \Theta(g(n))$ if $\exists c_1, c_2 > 0, n_0 > 0$ such that $\forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ i.e. $\Theta(g) = O(g) \cap \Omega(g)$
- $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \implies f(n) = o(g(n))$
- $\lim_{n \to \infty} \frac{f(n)}{g(n)} < \infty \implies f(n) = O(g(n))$
- $0 < \lim_{n \to \infty} \frac{f(n)}{g(n)} < \infty \implies f(n) = \Theta(g(n))$
- $\lim_{n \to \infty} \frac{f(n)}{g(n)} > 0 \implies f(n) = \Omega(g(n))$
- $\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty \implies f(n) = \omega(g(n))$

3 Recurrences

General form: $T(n) = aT(\frac{n}{b}) + f(n)$

- Telescoping: express in form $\frac{T(n)}{g(n)} = \frac{T(\frac{n}{b})}{g(\frac{n}{b})} + h(n)$
- Recursion tree: draw tree, sum each node (can sum over level first then over height)
- **Master Theorem**: $a \geq 1, b > 1, f$ asymptotically positive
  1. $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$. $T(n) = \Theta(n^{\log_b a})$
  2. $f(n) = \Theta(n^{\log_b a} \log^k n)$ for some $k \geq 0$. $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
  3. $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some $\epsilon > 0$ and satisfies **regularity condition** $af(\frac{n}{b}) \leq cf(n)$ for some $c < 1$. $T(n) = \Theta(f(n))$
- Substitution: guess and check by induction. Induction hypothesis: $c_1 n^k -$(lower order terms)

4 Divide and Conquer

Invariant: condition which is true at the start of every iteration. To show correctness check

- Initialisation: invariant is true at iteration 1
- Maintenance: if invariant is true for iteration n, it remains true for iteration $n + 1$
- Termination: when the algorithm ends, the invariant helps the proof of correctness

5 Randomisation

- Las Vegas algorithms: (1) output always correct (2) running time depends on random bits, with small probability that it may be large (3) expected running time is bounded by given time-bound function
- Monte Carlo algorithms: (1) answer may be incorrect with small probability (2) running time is always bounded by given time-bound function

6 Dynamic Programming

- Optimal substructure: optimal solution contains optimal solutoins to subproblems
- Overlapping subproblems: recursive solution contains a small number of distinct subproblems repeated many times
- Top-down saves computation of unnecessary subproblems but can suffer from overhead of recursive calls, bottom-up is the opposite