Link: Fall 2011 Rec 1 Notes

1 Asymptotic Complexity

1.1 Θ, O and Ω - CLRS Chapter 3

$f(n) = \Theta(g(n))$

- **Intuition:** $f \approx g$
- **Words:** After a certain point, we can put a lower and upper bound on f using constant multiples of g.
- **Math:** First, understand $\Theta(g(n))$ is actually a *s*et of functions, $a(n)$, st.

  $$\exists\, c_1, c_2, n_0 > 0 : c_1 g(n) \leq a(n) \leq c_2 g(n)\ \forall n \geq n_0$$

  So, $f(n) = \Theta(g(n))$ implies f is in this set, i.e.

  $$\exists\, c_1, c_2, n_0 > 0 : c_1 g(n) \leq f(n) \leq c_2 g(n)\ \forall n \geq n_0$$

- **Example:**

  $$x^2 + 1.5x(\sin(x)) + 6006 = \Theta(x^2)$$

  Picture in Fall 2011 Notes.

Repeat with however much depth for O and Ω.
Useful equations are

$$x(1 + \sin(x))$$

This has a lower bound of 0 but an upper bound $O(x)$.

$$(1 + \sin(x))x^2 + x$$

Lower bound of $\Omega(x)$ but upper bound of $O(x^2)$.
Each of these has no Θ bound because upper and lower bounds do not differ by constant factor.

1.2 Practice

1.2.1 Starters

$$10^{80} = O(1) \quad (1)$$
$$(20n)^7 = O(n^7) \quad (2)$$
$$(3)$$

1.2.2 Log Properties

$\log(n^{100}) = \Theta(\log(n))$ because $\log(n^{100}) = 100\log(n)$.

$\log_5(n) = \Theta(\log(n))$ because $\log_5(n) = \frac{\log(n)}{\log(5)}$

Mention that comparing logs of expressions is helpful in determining which is bigger in the limit.

1.2.3 Harder Log Problems

$$\log_{\log(5)}(\log(n)^{100}) = \Theta(\log(\log(n)) \tag{4}$$

$$\log_{\log(5)}(\log(n^{100})) = \Theta(\log(\log(n)) \qquad \text{but for different reasons} \tag{5}$$

1.2.4 Sterling's Approximation

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

$$\log\binom{n}{\frac{n}{2}} = \log\frac{n!}{\left(\frac{n}{2}\right)!\left(\frac{n}{2}\right)!} \tag{6}$$

$$= \log\frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{\left(\sqrt{2\pi n/2}\left(\frac{n}{2e}\right)^{n/2}\right)^2} \tag{7}$$

$$= \log\frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{\pi n\left(\frac{n}{2e}\right)^n} \tag{8}$$

$$= \log\left(2^n\sqrt{\frac{2}{\pi n}}\right) \tag{9}$$

$$= n\log(2) + 1/2\log\sqrt{2} - 1/2\log\pi n \tag{10}$$

$$= \Theta(n) \tag{11}$$

2 Peak Finding

Review recurrences of these algos!

If you want to have a harder recurrence try these from Fall 2011 Exam 1.

$$T(n) = T(n/3) + T(2n/3) + \Theta(n) \qquad \Theta(n\log n) \tag{12}$$

$$T(n) = \log n + T\left(\sqrt{n}\right) \qquad \Theta(\log n) \tag{13}$$

1. Review 1D algorithm

   Use induction on the recursion depth, showing that the algorithm returns a correct peak.

Base Case: If the algorithm returns with no recursive calls, then it checks directly that the return value is a peak.

Inductive Step: Suppose the algorithm splits on the midpoint location m and makes a recursive call. It could be to the left (lower indices) or the right (higher indices); suppose wlog that the call is to the right. Assume that the recursive call does what it is supposed to—returns a peak for the sub-array. The algorithm returns the result of the recursive call, so we must argue that it must also be a peak for the entire array.

That is immediate by definition of a peak if the peak occurs anywhere except for location $m + 1$. But if we find a peak at location $m + 1$ in the sub-array, then we are not quite done: to show that it is a peak in the larger array, we must also check that $v(m + 1) \geq v(m)$. But we know this is true because that was the condition that made the algorithm decided to recurse to the right.

2. Review 2D algorithm which cuts the array into quarters. Beware, it must also track best seen! See `crossalgo.py`.

3. Review circular peak finding algorithm as last problem in Spring 2012 PSET 1.

   Link: Spring 2012 PSET 1 Solutions