TCSS 343 - Week 1 - Monday

Jake McKenzie

January 13, 2019

Asymptotics with Mathematical Induction

"Programs must be written for people to read, and only incidentally for machines to execute".

...

Harold Abelson

"You shouldn't try optimizing something that you can't measure."

...

Elecia White

"People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world."

...

Pedro Domingos

Exact answers are nice when you can find them. Often times we can't or don't care to find them. As computer scientists we care about the behavior of the algorithms we employ. If we run into a recurrence relation or summation that expressed loosly the behaivor of these algorithms we'd like to know something about their runtime.

Asymptotic Notation in Seven Words: suppress constant factors and lower-order terms

Constant factors are things that will be language and system dependent while lower order terms are irrelevant for large inputs.

Now consider these definitions:

Big-O Notation

If $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} \to 0$ then $f(n) \in O(g(n))$

This is another way of saying that $f(n) \leqslant c \cdot g(n)$ for some position c.

Big-Ω Notation

If $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} \to \infty$ then $f(n) \in \Omega(g(n))$

This is another way of saying that $f(n) \geqslant c \cdot g(n)$ for some position c.

Big-Θ Notation

If $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} \to k$ where k is a positive finite number then $f(n) \in \Theta(g(n))$

This is another way of saying that $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$.

Now while we're in this course, we will typically need to show that these relationships are true when prompted, but it's always good to know that the following is true where $0 < \epsilon < 1 < c$.

$$1 < \log\log n < \log n < n^{\epsilon} < n^{c} < n^{logn} < c^{n} < n^{n} < c^{c^{n}}$$

This asymptotic pecking order above is from Don Knuth's Concrete Mathematics.

From Tim Roughgarden's Algorithms Illuminated:

$$\max\{f(n), g(n)\} \leqslant f(n) + g(n) \wedge 2 \cdot \max\{f(n), g(n)\} \geqslant f(n) + g(n)$$

(reminder that $\wedge$ is the logical "and" symbol.)

∴

0. Show the following by using the definition of big Theta and the information immediately above to show that $\max\{f(n), g(n)\} \in \Theta(f(n)+g(n))$

$\max\{f, g\} \leq f+g$ $\qquad$ $\max\{f, g\} \geq \frac{1}{2}(f+g)$

bring it all together

$$\frac{1}{2}(f+g) \leq \max\{f, g\} \leq f+g$$

$$\left.\begin{aligned} a &= \tfrac{1}{2} \\ b &= 1 \end{aligned}\right\} \therefore$$

$$\max\{f, g\} \in \Theta(f+g)$$

1. Arrange the following functions in order of increasing growth rate, with $g(n)$ following $f(n)$ in your list if and only if $f(n) \in O(g(n))$. That means using the limit definitions I gave you or by proof technique you've seen in class. You cannot simply use Don's asymptotic pecking order I stated but I strongly suggest you use the asymptotic pecking order as a guide to the neighborhood of a correct answer. If two function are tightly bounded together asymptotically then we can say they are $=_p$ to each other.

a) $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

b) n^2

c) $n^{0.99} \log n$

greater than 1

d) 1.01^n $\lim_{n\to\infty} \frac{2\cdot 2^n - 1}{1.01^n} = \lim_{n\to\infty} 2\left(\frac{2}{1.01}\right)^n - \frac{1}{1.01^n} \to \infty$

$\to\infty \qquad \to 0$

e) $4^{\log n} = (2^2)^{\log_2 n} = 2^{2\log_2 n} = 2^{\log_2 n^2} = n^2 \therefore b =_p e$

f) $100^{\sqrt{n}}$

g) $\log 2^{\frac{n}{2}} = \frac{n}{2}$

h) $10000n$ $h =_p \Theta$

$h =_p g < c < b =_p e < d < d < a$

2. Arrange the following functions in order of increasing growth rate, with $g(n)$ following $f(n)$ in your list if and only if $f(n) \in O(g(n))$. That means using the limit definitions I gave you or by proof technique you've seen in class. You cannot simply use Don's asymptotic pecking order I stated but I strongly suggest you use the asymptotic pecking order as a guide to the neighborhood of a correct answer. If two function are tightly bounded together asymptotically then we can say they are $=_p$ to each other.

a) $n^{\frac{5}{3}}$

b) $\sum_{i=0}^{n}(i+1) = \sum_{i=0}^{n} i + \sum_{i=0}^{n} 1 = \frac{n(n+1)}{2} + n - 0 + 1 \in \Theta(n^2)$

c) $n\sqrt{n} = n^{\frac{3}{2}}$

d) $\log n^n = n \log n$

e) $\frac{n}{\log n}$

f) $2^{\frac{n}{2}}$

g) $\log \log n$

h) $n^{1.5} = n^{\frac{3}{2}}$

$g < e < d < c =_p h < a < b < f <$

3. Prove by induction that: $\sum_{i=1}^{n} 3i - 2 = \frac{n(3n-1)}{2}$.

BC:

$$\sum_{i=1}^{1} 3i-2 = 3(1)-2 = 1$$

$$\frac{1(3(1)-1)}{2} = \frac{2}{2} = 1 \quad ✓$$

IH:

$$P(K): \sum_{i=1}^{K} 3i-2 = \frac{K(3K-1)}{2}$$

IS:

$$\sum_{i=1}^{K+1} = \sum_{i=1}^{K} 3i-2 + 3(K+1)-2 = \frac{(K+1)(3(K+1)-1)}{2}$$

$$\frac{K(3K-1)}{2} + 3K+1 = \frac{(K+1)(3K+2)}{2}$$

$$\frac{3K^2-K+6K+2}{2} = \frac{3K^2+5K+2}{2}$$

$$\frac{3K^2+5K+2}{2} = \frac{3K^2+5K+2}{2}$$

4. Show that for any real constant a and b, where $b > 0$:

$$(n + a)^b \in \Theta(n^b)$$

$$\lim_{n \to \infty} \frac{(n+a)^b}{n^b} = \lim_{n \to \infty} \frac{n^b(\overset{\nearrow 1}{1 + \frac{a}{n}})^b}{n^b} \to 1$$

▨

lower bound:

$$(n+a)^b \leq (n+|a|)^b, \text{ where } n > 0$$
$$\leq (n+n)^b, \ \forall\, n \geq |a|$$
$$= (2n)^b$$
$$= 2^b \cdot n^b \quad \therefore (n+a)^b \in \Omega(n^b)$$

upper bound:

$$(n+a)^b \geq (n-|a|)^b, \text{ where } n > 0$$
$$\geq (\tfrac{1}{2}n)^b \text{ where } n \geq 2|a|$$
$$\text{as } \tfrac{n}{2} \leq n - |a|, \text{ for } n \geq 2a \quad \therefore (n+a)^b \in O(n^b)$$

$$2^{-b}n^b \leq (n+a)^b \leq 2^b n^b$$
$$n_0 \geq 2|a|$$

▨

5. Analyze the following piece of pseudocode and give a tight (Θ) bound on the running time as a function of n. Show your work. A formal proof is not required, but you should justify your answer. The input of this function $n \in \mathbb{Z}^+$ (positive integers). I suggest expressing this function as a summation.

```
for $i \leftarrow 1$ to 2018 do
    $chimp \leftarrow n^4$
    while $chimp > 1$
        $chimp \leftarrow chimp/2$
```

$$\sum_{i=1}^{2018} c\log_2 n^4 = 4c\log_2 n \sum_{i=1}^{2018} 1$$

$$= 4c\log_2 n\,(2018-1+1)$$

$$= 8072c\log_2 n$$

$$\lim_{n\to\infty} \frac{8072c\log_2 n}{\log_2 n} \rightarrow \boxed{8072c}$$

$$\boxed{8072c\log_2 n \in \Theta(\log n)}$$