

Problem Set 1

Max Harris

January 16, 2014

Collaboration: Attended tail-end of office hours on Wednesday.

1. (3 pts each) For each of the following claims, determine whether they are true or false.

(a) $n + 3 = O(n^3)$

True. $n + 3$ is bounded by cn^3 from above for $c \geq 2$ for example. We can verify this by checking $\lim_{n \rightarrow \infty} \frac{n+3}{n^3} = \lim_{n \rightarrow \infty} \frac{1}{3n^2} = 0$. This shows that n^3 grows faster than $n + 3$ for all nonnegative n .

(b) $3^{2n} = O(3^n)$

False. Note that $3^{2n} = 3^n \times 3^n$. When we compare the rate of growth of $3^n \times 3^n$ to 3^n by checking $\lim_{n \rightarrow \infty} \frac{3^n \times 3^n}{3^n} = \lim_{n \rightarrow \infty} \frac{3^n}{1} = \infty$ we see that the top term, 3^{2n} grows faster and is therefore not bounded from above by 3^n .

(c) $n^n = o(n!)$

False. Let's compare n^n with $cn! \mid c \geq 0$. For this to be true we would want to show that for every c , there exists some n_0 such that for all $n > n_0, n^n < cn!$. We can take $\lim_{n \rightarrow \infty} \frac{n^n}{n!} = \infty$ which shows that n^n grows faster than $n!$. Now for any c we take, for example $c = 1$, since n^n grows faster than $n!$, n^n will never be strictly less than $n!$ for any $n \geq 0$.

(d) $\frac{1}{3n} = o(1)$

True. Showing that $\lim_{n \rightarrow \infty} \frac{1/(3n)}{1} = \lim_{n \rightarrow \infty} \frac{1}{3n} = 0$ means that $1/(3n)$ is bounded from above by 1 and converging to 0 for all $n > \frac{1}{3}$, thus for any c , there will be an n_0 such that for all $n > n_0, 1/(3n) < c$.

(e) $\ln^3 n = \Theta(\lg^3 n)$ \lg means \log_2

I really am not sure how to do this. \log_2 confuses me.

2. (4 pts each) Simplify the following expressions.

(a) $\frac{d}{dt}(3t^4 + \frac{1}{3}t^3 - 7)$

$$\frac{d}{dt}(3t^4 + \frac{1}{3}t^3 - 7) = (4 \times 3)t^3 + \frac{3}{3}t^2 = 12t^3 + t^2$$

(b) $\sum_{i=0}^k 2^i$

$2^0 = 1$ and $2^1 = 2$ which is one more than the previous term. $2^2 = 4$ which is 1 more

than $2 + 1$. $2^3 = 8$, which is 1 more than $2^2 + 2^1 + 1$. Following this trend it seems as though $2^{n+1} = \sum_{i=0}^n 2^i + 1$ or $2^{n+1} - 1 = \sum_{i=0}^n 2^i$.

(c) $\Theta\left(\sum_{k=1}^n 1/k\right)$

I don't know because even if I figured out the summation, I don't understand what it's asking by putting a Θ in front.

3. (23 pts) T is a balanced binary search tree storing n values. Describe an $O(n)$ -time algorithm that takes input T and returns an array containing the same values in *ascending* order.

The key to this is that the tree is balanced. This allows us to create an array, then traverse the tree with an *in order* traversal, visiting each node only once and inserting the value at the visited node into the array, thereby getting an array containing the values in ascending order in $O(n)$ -time.

4. (10 pts each) Acme Corp. has asked Professor Flitwick to develop a faster algorithm for their core business. The current algorithm runs in $f(n)$ time. (For concreteness, assume it takes $f(n)$ microseconds to solve a problem of size exactly n.) Flitwick believes he can develop a faster algorithm, which takes only $g(n)$ time, but developing it will take t days. Acme only needs to solve a problem of size n once. Should Acme pay Flitwick to develop the faster algorithm or should they stick with their current algorithm? Explain.

- (a) Let $n = 41$, $f(n) = 1.99^n$, $g(n) = n^3$ and $t = 17$ days.

Using Google to convert 17 days to microseconds, we get $1.47e+12$, which seems close to $1.99^{41} = 1.79e+12$, but not close enough to make $41^3 = 68921$ significant. Because one increases polynomially and the other exponentially, Acme can save 3.72 days by choosing Professor Flitwick to create his $g(n)$ algorithm.

- (b) Let $n = 10^6$, $f(n) = n^{2.00}$, $g(n) = n^{1.99}$ and $t = 2$ days.

After waiting 2 days, or $17.28e+10$ microseconds, Flitwick's algorithm $g(n)$ would add an additional $(10^6)^{1.99} = 86.10e+10$ giving us $1.043e+12$, whereas if Acme continues to use the old algorithm $f(n)$, we get a total of $(10^6)^2 = 1e+12$. Comparing the difference, acme saves about 12 hours by continuing to use their old algorithm $f(n)$.

5. (5 pts each) Using the mathematical definition of Big-O, answer the following. Show your work.

- (a) Is $2^{nk} = O(2^n)$ for $k > 1$?

No. When $k > 1$, say $k = t$, then $2^{tn} = 2^n \times 2^n \times \dots \times 2^n$ (t times). Thus when compared to 2^n by taking $\lim_{n \rightarrow \infty} \frac{2^{kn}}{2^n} = \lim_{n \rightarrow \infty} \frac{2^{(k-1)n}}{1} = \infty$ when $k > 2$.

- (b) Is $2^{n+k} = O(2^n)$, for $k = O(1)$?

Yes, When $k = O(1)$, k is constant. Thus, $2^{n+k} = 2^n \times 2^k$ and since 2^k is constant, and easier way to look at this is to let $c = 2^k$ so we can ask if $c2^n = O(2^n)$. Since we know constants don't effect the bounding order, we can conclude that $c2^n = 2^{n+k} = O(2^n)$ is true.

6. (20 pts) Is an array that is in sorted order also a min-heap? Justify.

First, the array must be sorted in ascending order to be a min-heap. Take for example the array containing the integers 1-5 and we want to make a binary min-heap. The root would be 1, 1's children would be 2 and 3, and 2's children would be 4 and 5. We can see that this would be a valid min-heap. It doesn't have to be binary either. You can imagine adding any number children to the parent nodes and as long as it is coming from an array sorted in ascending order, we will produce a valid min-heap.

Now, it is important to mention that going from an array in ascending order to a heap produces a valid min-heap without making any comparisons. However, when we move from a min-heap, which only guarantees that nodes will be greater than their parent nodes, to a sorted array, we will need to compare child nodes with other child nodes before sorting them into the array.

For all intents and purposes however and to concretely answer the question at hand, an array that is in sorted order is *not* necessarily a min-heap, unless of course we can guarantee that the array is sorted in ascending order.