CS 624 - HW 1

Max Weiss

3 February 2020

1 (a) A derivation on p20 of the handout gives

$$\sum_{n=0}^{\infty} f_n x^n = \frac{x}{1-x-x^2}$$

Differentiating both sides,

$$\sum_{n=0}^{\infty} n f_n x^{n-1} = \frac{(1-x-x^2)-x(-1-2x)}{(1-x-x^2)^2} = \frac{1+x^2}{(1-x-x^2)^2}. \tag{1}$$

Substituting 1/2 for x,

$$\sum_{n=0}^{\infty} \frac{n f_n}{2^{n-1}} = \frac{1+\frac{1}{4}}{(1-\frac{1}{2}-\frac{1}{4})^2} = \frac{\frac{5}{4}}{\frac{1}{16}} = 20.$$

1 (b) Substituting 1 for x in (1) would appear to give

$$\sum_{n=0}^{\infty} n f_n = \frac{(1-1-1)-1(-1-2)}{(1-1-1)^2} = 2. \tag{2}$$

However, (2) is false; indeed, the sum on the left hand side does not converge at all. For, toward a contradiction, suppose that the sum converges to d. Since $f_n > n$ for all $n > 3$ (as is clear by induction on n), it follows that $nf_n > n$ for all $n > 3$. Thus, $\sum_{n=0}^{e} nf_n > d+1$ for any e greater than both d and 3. So, any partial sum of length at least e cannot come within even $\epsilon = 1$ of d, contradicting the definition of convergence.

2(a) According to the binomial theorem,

$$(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k. \tag{3}$$

2(b) Similarly,

$$(1-x)^n = \sum_{k=0}^{n} \binom{n}{k} (-x)^k. \tag{4}$$

2(c) Using (3) and (4) above,

$$\begin{aligned}(1+x)^n + (1-x)^n &= \sum_{k=0}^{n} \binom{n}{k} x^k + \sum_{k=0}^{n} \binom{n}{k} (-x)^k \\ &= \sum_{k=0}^{n} \binom{n}{k} (x^k + (-x)^k) \\ &= 2\left(\binom{n}{0}x^0 + \binom{n}{2}x^2 + \binom{n}{4}x^4 + \cdots\right).\end{aligned}$$

2(d) Likewise using (3) and (4) above,

$$\begin{aligned}(1+x)^n - (1-x)^n &= \sum_{k=0}^{n} \binom{n}{k} x^k - \sum_{k=0}^{n} \binom{n}{k} (-x)^k \\ &= \sum_{k=0}^{n} \binom{n}{k} (x^k - (-x)^k) \\ &= 2\left(\binom{n}{1}x^1 + \binom{n}{3}x^3 + \binom{n}{5}x^5 + \cdots\right). \qquad (5)\end{aligned}$$

2(e) Setting $x = 1$, the equation (5) implies

$$2^n = (1+1)^n - (1-1)^n = 2\left(\binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots\right)$$

and so

$$\binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots = 2^{n-1}$$

which should be simple enough for the econ prof!

3 (a) Let me write $f(n) = n^2$ and $g(n) = 2^n$. Note that

$$f(n+1) = (n+1)^2 = n^2 + 2n + 1 = f(n) + 2n + 1$$

while

$$g(n+1) = 2^{n+1} = g(n) + g(n).$$

It follows that if $g(n) > f(n)$ and $g(n) > 2n+1$, then $g(n+1) > f(n+1)$.

I'll argue that $g(n) > 2n+1$ for all $n \geq 5$. This is clear for $n = 5$. Now assume that $g(n) > 2n+1$. Then

$$g(n+1) = 2g(n) > 2(2n+1) > 2(n+1) + 1$$

and so the claim follows by induction.

The above two claims imply that if $g(n) > f(n)$, then $g(n+1) > f(n+1)$ for all $n > 4$. However, $g(5) > f(5)$. So if $x = 5$, then $f(n) \leq g(n)$ for all $n \geq x$. So $f \in O(g)$, as desired.

3 (b) Note that $2^{n+1} = 2 * (2^n)$. So if $c = 2$ and $x = 0$, it follows that that $2^{n+1} \leq c * 2^n$ for all $n \geq x$.

3 (c) Suppose there were c, x such that $2^{2n} \leq c * 2^n$ for all $n \geq x$. Since $2^n > 0$, it follows from $2^{2(n+1)} \leq c * 2^{n+1}$ that $2^{n+1} \leq c$ for all $n \geq x$.

However, $2^n > n$ for all $n \geq 0$ (as is readily proved by induction on n). Hence, $2^{c+1} > c + 1 > c$. We must therefore have $c < x$, so that $c < x < 2^x < 2^{x+1}$, a contradiction.

3 (d) Let $f(n) = 2n$, and let $g(n) = n$. Clearly $f \in O(g)$, since we may pick $c = 2$. But, writing $\hat{f}(n) = 2^{f(n)}$ and $\hat{g}(n) = 2^{g(n)}$, the above response to exercise 3c says precisely that $\hat{f} \notin O(\hat{g})$.

4 First, note that $a^{\log_a(x)} = x$, so that

$$\log_a(x) \log_b(a) = \log_b(a^{\log_a(x)}) = \log_b(x). \tag{6}$$

Therefore

$$\log_a(x) = \log_b(x) / \log_b(a) \tag{7}$$

(the "change of base formula"). So setting $c = 1/\log_b(a)$ and $x = 1$, it follows that $\log_a(x_0) \leq c \log_b(x_0)$ for all $x_0 > x$.

5 (a) I'll argue by induction on n. Clearly $x^j \in O(x^n)$ if $n = j$. Now, suppose that $x^j \leq cx^n$ for all x. Picking $\hat{x}$ to be the greater of x and 1, it follows that $x^j \leq cx^n \leq cx^{n+1}$ for all $x \geq \hat{x}$, as desired.

Before continuing to (b), note that if $g \in O(f)$, then $O(g) \subseteq O(f)$. Hence part (a) implies that $O(x^j) \subseteq O(x^n)$ for all $j \leq n$.

(b) I'll argue by induction on n. The claim is trivial for $n = 1$. Writing $p(x) = \sum_{i=1}^{n} a_n x^n$, suppose that $\sum_{i=1}^{n} a_n x^n \in O(x^n)$. It then follows from the above remark on part (a) that $p(x) \in O(x^{n+1})$. Meanwhile, clearly $a_{n+1}x^{n+1} \in O(x^{n+1})$. The claim follows from the fact that $O(x^{n+1})$ is closed under addition (Lemma 3.1 in the handout).

It now suffices to show that $O(f)$ is closed under addition for any f. Well if $O(f)$ contains both g and h, then there are x_0, x_1 and c_0, c_1 such that $g(n) \leq c_0 f(n)$ and $h(n) \leq c_1 f(n)$ for all n respectively no less than x_0, x_1. Picking c and x as $c_0 + c_1$ and the larger of x_0, x_1, it follows that $g(x) + h(x) \leq cf(n)$ for all $n \geq x$.

(c) First, note that $p(x) \geq a_n x^n$ for all $x > 0$, so it suffices to show that $a_n x^n \in \Omega(x^n)$. Since $a_n > 0$, we may simply pick $c = a_n$ to affirm that $a_n x^n \geq cx^n$ for all x so that $a_n x^n \in \Omega(x^n)$ as desired.

(d) Suppose that $p(x) \in O(x^d)$, where $p(x) = \sum_{i=1}^{n} a_i x^i$. Toward a contradiction further suppose that $a_n > 0$. Then likewise $a_n x^n \in O(x^d)$, and so, since $a_n > 0$, also $x^n \in O(x^d)$. Then there are $c, \hat{x}$ such that $x^n \leq cx^d$ for all $x \geq \hat{x}$. So writing $m = n - d$, we have $m > 1$ while $x^m \leq c$ for all $x \geq \hat{x}$. This contradicts the fact that $x^m > c$ for any x greater than both 1 and c.

6 Suppose that $f \in O(g)$ and $g \in O(h)$. Then there are c_0, c_1, x_0, x_1 such that $f(x) \leq c_0 g(x)$ and $g(x) \leq c_1 h(x)$ for all x greater than x_0, x_1 respectively. Picking c to be $c_0 c_1$ and $\hat{x}$ to be the larger of x_0, x_1, it follows that $f(x) \leq ch(x)$ for all $x \geq \hat{x}$.

7 (a) $T(n) = 2T(\frac{n}{2}) + n^4$. Setting $a = 2$ and $b = 2$, it follows that $\log_b a = 1$ so that $f(n) = n^4 \in O(n^{\log_b a + \epsilon})$ for $\epsilon = 3$. Furthermore, picking $c = 1$,

$$af\left(\frac{n}{b}\right) = 2\left(\frac{n}{2}\right)^4 = \frac{n^4}{8} \leq n^4 = cf(n)$$

for all sufficiently large n (e.g., > 1).

So, we are in case (3) of the master theorem, and $T(n) = \Theta(f(n)) = \Theta(n^4)$.

(b) $T(n) = T(\frac{7n}{10}) + n$. Setting $a = 1$ and $b = \frac{10}{7}$, it follows that $\log_b a = \log_{\frac{10}{7}} 1 = 0$. So here $f(n) = n^1 \in O(n^{\log_b a + \epsilon})$ for $\epsilon = 1$. Furthermore, again picking $c = 1$,

$$f\left(\frac{7n}{10}\right) = \frac{7n}{10} \leq n = cf(n)$$

for all $n \geq 0$.

So we are again in case (3) of the master theorem, and

$$T(n) = \Theta(f(n)) = \Theta(n).$$

(c) $T(n) = 16T(\frac{n}{4}) + n^2$. With $a = 16$ and $b = 4$, it follows that $f(n) = n^2 \in O(n^2) = O(n^{\log_b a})$.

Here we are in case (2) of the master theorem, and

$$T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(n^2 \lg n).$$

(f) $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$. With $a = 2$ $b = 4$, it follows that $f(n) = \sqrt{n} = n^{\frac{1}{2}} \in O(n^{\frac{1}{2}}) = O(n^{\log_b a})$.

So again we are in case (2) of the master theorem, and

$$T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(\sqrt{n} \lg n).$$

(g) $T(n) = T(n-2) + n^2$. In this case, the recurrence doesn't (apparently) have a form suited to the master theorem. However, note for example that

$$T(7) = T(1) + 3^2 + 5^2 + 7^2$$

and this clearly generalizes: $T(n)$ is the sum of a constant plus the squares of all odd numbers from 3 up to n. Geometrically, the sum can be approximated as a pyramid with with base lengths and height each in some constant ratio to n. So a reasonable guess is that $T(n) \in \Theta(n^3)$.

Following the substitution method, I'll now argue first, by induction on n, that $T(n) \geq \frac{1}{6}n^3$ for all $n \geq 3$. For the basis step, we need to consider both $n = 3$ and $n = 4$, which amounts to observing that $3^2 + T(1) \geq \frac{1}{6}3^3$ and that $4^2 + T(1) \geq \frac{1}{6}4^3$. Assuming now that $T(m) \geq \frac{1}{6}m^3$ for all $m < n$, it follows for $n > 4$ that

$$\begin{aligned} T(n) &= T(n-2) + n^2 \\ &\geq \frac{1}{6}(n-2)^3 + n^2 \\ &= \frac{1}{6}n^3 + 2n - \frac{4}{3} \\ &\geq \frac{1}{6}n^3. \end{aligned}$$

It remains to find conversely an bound on T. For simplicity, let's assume that the times $T(1)$ and $T(2)$ are negligible (since T is unbounded, it will eventually swamp whatever constants they may be). I'll argue, again by induction on n, that $T(n) \leq n^3$ for all $n \geq 12$. It's easy to verify that this holds for $n = 12$ and $n = 13$, so suppose that it holds for all $m < n$. It follows, certainly for $n > 13$, that

$$\begin{aligned} T(n) &= T(n-2) + n^2 \\ &\leq n^3 - 6n^2 + 12n - 8 + n^2 \\ &\leq n^3. \end{aligned}$$

8. [See attachment.]

9 (a) There exist nonempty sets of points in the plane which contain no uniquely closest pair. For example, $\{(0,1), (0,2), (0,3)\}$.

(b) The algorithm takes time proportional to n^2, for all n. To see this, note that the collection of pairs of elements of an n-element set has size proportional to n^2. (If the pairs are ordered pairs, it is exactly n^2, else it is $\frac{n(n+1)}{2}$). Since the algorithm measures a distance for each pair, it must take at least $\sim n^2$ steps. Conversely, it requires only a constant number of tasks per pair; besides measuring the distance d, it must compare d with the current maximum distance m and in the worst case replace m with d.

10. Suppose that the nodes of a binary tree are labelled with the entries of an array A. Specifically, the node of depth 0 is labelled with the first entry $A[1]$, the nodes of depth 1 with the second and third entries $A[2]$ and $A[3]$, and so on.[1] The statement in question is this:

(*) If N is labelled with $A[n]$, then the children of N are labelled with $A[2n]$ and $A[2n+1]$.

This statement has counterexamples if the array entries are not all distinct (for example, suppose that the root and its children are labeled 0, but the children's children are in turn labeled 1; then the root is labeled with $0 = A[2]$, but its children are not labeled with $A[4]$ or $A[4+1]$).

So instead, I will begin by proving a special case of (*), namely where $A[i] = i$ for all i. From this I will derive a weaker version of (*).

Let's say that the index of a given node is the array index of the unique entry by which it is labelled. The special case to handled is now this:

(A) Suppose that a given node N has index n, and that each node of index at most n has two children. Then the children of N have indices $2n$ and $2n+1$.

Toward a proof of (A), first note two facts. In both cases, assume that the depth d is less than the depth of N (so that "all" nodes of depth d exist).

(B) The number of nodes of depth d is 2^d. This is clear for $d = 0$. As for $d+1$, by induction there are 2^d nodes of depth d, and the nodes of depth $d+1$ are precisely the children of a node of depth d. Since each node has two children, and no two nodes have a child in common, it follows that there are $2(2^d) = 2^{d+1}$ nodes of depth $d+1$.

(C) The number of all nodes of depth less than d is $2^d - 1$. This is vacuous for $d = 0$. Now, suppose the claim holds for d. The nodes of depth less than $d+1$ are precisely the nodes of depth less than d, plus the nodes of depth d itself. By fact (B), the number of nodes of depth d is 2^d, while by induction hypothesis, the number of nodes of depth less than d is $2^d - 1$. However, $2^d + 2^d - 1 = 2^{d+1} - 1$, which proves the claim.

Now, suppose that the node N is the ith node of depth d. By (C) above, there are $2^d - 1$ nodes of depth less than d. So, N has index $2^d + i - 1$. On the other hand, N has $i - 1$ predecessors at depth d, and every predecessor of N has two children. So, the children of N have $2(i-1)$ and $2(i-1)+1$ predecessors at depth $d+1$. Again using fact (C), it follows that they have indices $2^{d+1} - 1 + 2(i-1) + 1 = 2(2^d + i - 1) = 2n$ and $2^{d+1} - 1 + 2(i-1) + 2 = 2n+1$ respectively. This completes the proof of (A).

Finally, let's return to the situation where the nodes are labeled with values of an arbitrary array A. Suppose that N is labeled with $A[m]$, and that n is the

[1] I'm just assuming the fact that there always exists a (unique) such labelling.

index of N. By construction of the labeling, we must then have $A[m] = A[n]$. So, by (A) we get the generalization I promised earlier:

(D) If node N is labeled with $A[n]$, and if all predecessors of N have two children, then the children of N are labeled with $A[2n']$ and $A[2n' + 1]$ for some n' such that $A[n'] = A[n]$. xf