18-799 Algorithms and Computation in Signal Processing
Spring 2005
Assignment 1 - Solution

1. *(9 pts)* Show that the following identities hold by determining the explicit constants c and n_0 that are a part of the definition of O.

   First look up the definition of $O(f(n))$.

   (a) $n+1 = O(n)$
   Solution: It is, for all $n \geq 1$,
   $$n+1 \leq n+n = 2n.$$
   Thus, for $c = 2, n_0 = 1$, we have $n+1 \leq cn$ for all $n \geq n_0$.

   (b) $n^3 + an^2 + bn + c = O(n^3)$
   Solution: The trick is to get rid of all lower terms by converting them into multiples of n^3.
   For all $n \geq 1$:
   $$\begin{aligned} n^3 + an^2 + bn + c &\leq n^3 + |a|n^2 + |b|n + |c|n \\ &\leq n^3 + |a|n^3 + |b|n^3 + |c|n^3 \\ &\leq (1+|a|+|b|+|c|)n^3. \end{aligned}$$
   Therefore, we can choose $c = (1+|a|+|b|+|c|),\ n_0 = 1$ in the defintion of O. (Note that this c is different from the c in the polynomial.)

   (c) $n^5 = O(n^{\log_2 n})$
   Solution: Since $5 \leq \log_2(n)$, for $n \geq 32$, we have for $n \geq 32$,
   $$n^5 \leq n^{\log_2(n)}.$$
   Hence, we can choose $c = 1,\ n_0 = 32$.

2. *(16 pts)*

   (i) In the first class, you learned that $\Theta(\log_a n) = \Theta(\log_b n)$ for $a, b > 1$. Does $\Theta(a^n) = \Theta(b^n)$ hold? Justify your answer.
   Solution: This is true only if $a = b$. Otherwise, let's assume $a > b$. We show that $a^n \neq O(b^n)$ through a proof by contradiction.
   Namely, assume $a^n = O(b^n)$. Then, by definition of O, there is a constant c and n_0 such that for all $n \geq n_0$:
   $$a^n \leq cb^n.$$
   This implies (by applying the base-a log on both sides),
   $$n \leq b \log_a b + \log_a c.$$
   Since $a > b$ and thus $\log_a b < 1$, we can solve for n as
   $$n \leq \log_a c/(1 - \log_a b), \quad \text{for all} n \geq n_0,$$
   which is a contradiction (it does not hold since n grows to infinity); thus, the original assumption is wrong and we have $a^n \neq O(b^n)$ as desired.

   (ii) Prove or disprove: $2^{2n} = O(2^n)$.
   Solution: $2^{2n} = 4^n$. Above in (i) we showed already t hat $a^n \neq O(b^n)$, if $a > b > 1$.
   Other argument (informal): $2^{2n} = (2^n)^2$. This means that there cannot exist a constant c such that, for $n \geq n_0$, $(2^n)^2 < c2^n$, since this would imply $x^2 < cx$ for $x > n_0$.

(iii) Show that for $k > 0,\ \alpha > 1$: $n^k = O(\alpha^n)$ (i.e., polynomial functions grow slower than exponential functions).

Solution: We consider $\lim_{x\to\infty} x^k/\alpha^x$ and apply L'Hospital's rule

http://mathworld.wolfram.com/LHospitalsRule.html

k times. Remember that the derivative of α^x is $\log_e(\alpha)\alpha^x$:

$$\lim_{x\to\infty} \frac{x^k}{\alpha^x} = \lim_{x\to\infty} \frac{kx^{k-1}}{\log_e(\alpha)\alpha^x} = \dots = \lim_{x\to\infty} \frac{k!}{(\log_e(\alpha))^k\alpha^x} = 0.$$

This means, if we choose any $c > 0$, then there is an n_0 such that

$$n^k/\alpha^k < c,$$

which implies $n^k < c\alpha^k$ as desired.

(iv) Find a function $f(n)$ such that $f(n) = O(1)$, $f(n) > 0$ for all n, and $f(n) \neq \Theta(1)$. Justify the answer.

Solution: $f(n) = 1/n$. Obviously $0 < 1/n < 1$ for $n \geq 1$ and thus $1/n = O(1)$. Assume now that also $1/n = \Omega(1)$ and show that this leads to a contradiction. $1/n = \Omega(1)$ means that there is a constant $c > 0$ and n_0 such that

$$c \cdot 1 \leq 1/n, \quad \text{for } n \geq n_0,$$

which is obviously wrong (whenever $n > 1/c$).

Not easy to find an algorithm with $1/n$ as cost function :-).

3. *(21 pts)* Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible. Justify your answers.

(a) $T(n) = 2T(n/2) + n^3$. $a = 2, b = 2, f(n) = n^3$. This is Case 3. $T(n) = \Theta(n^3)$.

(b) $T(n) = T(9n/10) + n$. $a = 1, b = 10/9, f(n) = n$. This is Case 3. $T(n) = \Theta(n)$.

(c) $T(n) = 16T(n/4) + n^2$. $a = 16, b = 4, f(n) = n^2$. This is Case 2. $T(n) = \Theta(n^2 \log n)$.

(d) $T(n) = 7T(n/3) + n^2$. $a = 7, b = 3, f(n) = n^2$. This is Case 3. $T(n) = \Theta(n^2)$.

(e) $T(n) = 7T(n/2) + n^2$. $a = 7, b = 2, f(n) = n^2$. This is Case 1. $T(n) = \Theta(n^{\log_2 7})$.

(f) $T(n) = 2T(n/4) + \sqrt{n}$. $a = 2, b = 4, f(n) = \sqrt{n}$. This is Case 2. $T(n) = \Theta(\sqrt{n}\log n)$.

(g) $T(n) = 4T(n/2) + n^2 \log n$. This problem does not fall into any of the cases (recognizing this gives all points). One can solve it, e.g., by unrolling the recurrence (i.e., replacing $T(n/2)$ again by the recurrence etc.), assuming $n = 2^k$, and gets $T(n) = \Theta(n^2 \log^2(n))$ (gives extra points).

4. *(24 pts)* Compute the exact (arithmetic) cost

$$C(n) = (\text{number of adds}, \text{number of mults})$$

of the Karatsuba algorithm, recursively applied, for the multiplication of the polynomials:

$$h(x) = h_{n-1}x^{n-1} + \dots + h_0, \quad p(x) = p_{n-1}x^{n-1} + \dots + p_0,$$

assuming $n = 2^k$. (Solution: in written form at the end)

Extension to 4 (Extra Credit Problem, *20 pts*): Now compute the exact (arithmetic) cost

$$C(m, n) = (\text{number of adds}, \text{number of mults})$$

in the more general case

$$h(x) = h_{m-1}x^{m-1} + \dots + h_0, \quad p(x) = p_{n-1}x^{n-1} + \dots + p_0,$$

assuming $n = 2^k,\ m = 2^\ell,\ m \leq n$.

(Solution: in written form at the end)

5. *(30 pts)* Solve the recurrence $f_0 = 1,\ f_1 = 1, \quad f_n = f_{n-1} + 2f_{n-2}$, using the method of generating functions.

Solution: Our generating function is:

$$F(x) = \sum_{n>=0} f_n x^n$$

Step 1: $\sum f_n x^n = \sum f_{n-1} x^n + 2 \sum f_{n-2} x^n$

Step 2: $\sum_{n>=2} f_n x^n = \sum_{n>=2} f_{n-1} x^n + 2 \sum_{n>=2} f_{n-2} x^n$

Step 3:

$$\begin{aligned}
F(x) &= f_0 + f_1 + \sum_{n>=2} f_n x^n \\
F(x) &= 1 + x + \sum_{n>=2} f_{n-1} x^n + 2 \sum_{n>=2} f_{n-2} x^n \\
F(x) &= 1 + x + x \sum_{n>=2} f_{n-1} x^{n-1} + 2x^2 \sum_{n>=2} f_{n-2} x^{n-2} \\
F(x) &= 1 + x + x \sum_{k>=1} f_k x^k + 2x^2 \sum_{k>=0} f_k x^k \\
F(x) &= 1 + x + x \sum_{k>=0} f_k x^k + 2x^2 \sum_{k>=0} f_k x^k - x f_0 x^0 \\
F(x) &= 1 + x + xF(x) + 2x^2 F(x)
\end{aligned}$$

Step 4:

$$\begin{aligned}
F(x) &= 1 + xF(x) + 2x^2 F(x) \\
F(x) &= 1/(1 - x - 2x^2) = 1/(1+x)(1-2x)
\end{aligned}$$

Step 5:

$$\begin{aligned}
F(x) &= A/(1+x) + B/(1-2x) \\
A(1-2x) + B(1+x) &= 1 \\
A + B &= 1 \\
-2A + B &= 0 \\
\text{Solution: } A = 1/3, B &= 2/3
\end{aligned}$$

Step 6: $F(x) = 1/3 \sum_{n>=0} (-1)^n x^n + 2/3 \sum_{n>=0} 2^n x^n$

Step 7: $f_n = \frac{1}{3}(-1)^n + \frac{2}{3} 2^n = \frac{1}{3}(2^{n+1} + (-1)^n)$

4.) $h(x) = h_1(x^2) + x h_2(x^2)$ $\quad \deg(h_1) = \deg(p_1)$

$p(x) = p_1(x^2) + x p_2(x^2)$ $\quad = \deg(h_2) = \deg(p_2) = \frac{n}{2} - 1$

Note: adding 2 polynomials of degree k (both) requires $\leq k+1$ adds.

Karatsuba:

$$hp = \underbrace{h_1 p_1}_{\substack{\deg = n-2 \\ (\text{in } x^2)}} + [\underbrace{(h_1+h_2)}_{n/2}\underbrace{(p_1+p_2)}_{n/2} \underbrace{-}_{n-1} h_1 p_1 \underbrace{-}_{n-1} h_2 p_2] x + \underbrace{h_2 p_2 x^2}_{\substack{\deg = n-1 \\ (\text{in } x^2)}}$$

adds A_n: $n/2$ $\quad n/2$ $\quad n-1$ $\quad n-1$ $\quad n-2$ adds (solution below)

$\Rightarrow A_n = 4n - 4 + 3 A_{n/2}$, $A_1 = 0$

mults M_n: $M_1 = 1$, $M_n = 3 M_{n/2}$

$\xrightarrow[(n=2^k,\, N_k = M_n)]{\text{translate}}$ $N_0 = 1$, $N_k = 3 N_{k-1}$

$\xrightarrow{\text{solve}}$ $N_k = 3^k$

$\xrightarrow[\text{back}]{\text{translate}}$ $M_n = n^{\log_2 3}$

$$\boxed{\Rightarrow C_n = (4n-4,\ n^{\log_2 3})}$$

4n - 4 is wrong here, the right answer is below

$A_n = 3 A_{n/2} + 4n - 4$, $A_n = 0$

(translate $n = 2^k$

$A'_k = 3 A'_{k-1} + 4 \cdot 2^k - 4$

(formula from class

$$= \sum_{i=0}^{k-1} 3^i (4 \cdot 2^{k-i} - 4) = 4 \cdot 2^k \sum_{i=0}^{k-1} \left(\frac{3}{2}\right)^i - 4 \sum_{i=0}^{k-1} 3^i$$

$$= 4 \cdot 2^k \frac{\left(\frac{3}{2}\right)^k - 1}{1/2} - 4 \frac{3^k - 1}{2}$$

$$= 6 \cdot 3^k - 8 \cdot 2^k + 2$$

(translate

$$A_n = 6 \cdot n^{\log_2 3} - 8n + 2$$

4 extra.) $\deg h = m-1 = 2^{\ell}-1$ $h(x) = h_1(x^2) + x\, h_2(x^2)$

<u>$n \geq m$</u> $\deg p = n-1 = 2^{k}-1$ $p(x) = p_1(x^2) + x\, p_2(x^2)$

Karatsuba:
degrees:

$$hp = \underbrace{h_1 p_1}_{\frac{m}{2}-1 + \frac{n}{2}-1} + \Big[\underbrace{\underbrace{(h_1+h_2)}_{m/2-1}\underbrace{(p_1+p_2)}_{n/2-1}}_{m/2+n/2-2} - \underbrace{h_1 p_1}_{m/2-1+n/2-1} - \underbrace{h_2 p_2}_{m/2-1+n/2-1}\Big]x + \underbrace{h_2 p_2 x^2}_{m/2-1+n/2-1+1}$$

count adds: $m/2$, $n/2$, $m/2+n/2-1$, $m/2+n/2-1$, $m/2+n/2-2$

$$\Rightarrow \quad A_{m,n} = 3\, A_{m/2,\,n/2} + 2m + 2n - 4, \quad m \geq 2$$

$$A_{1,n} = 0$$

$m = 2^{\ell}$, $n = 2^{k}$:

$$A'_{\ell,k} = 3\, A'_{\ell-1,k-1} + 2^{\ell+1} + 2^{k+1} - 4, \quad \ell \geq 1$$

$$A'_{0,k} = 0$$

unroll: $A'_{\ell,k} = 3\left(3 A'_{\ell-2,k-2} + 2^{\ell} + 2^{k} - 4\right) + 2^{\ell+1} + 2^{k+1} - 4$

$$\cdots$$

$$= \sum_{i=0}^{\ell-1} 3^i \left(2^{k+1-i} + 2^{\ell+1-i} - 4\right)$$

$$= 2^{k+1}\sum_{i=0}^{\ell-1}\left(\tfrac{3}{2}\right)^i + 2^{\ell+1}\sum_{i=0}^{\ell-1}\left(\tfrac{3}{2}\right)^i - 4\sum_{i=0}^{\ell-1} 3^i$$

$$\cdots$$

$$= 4\cdot 2^{k-\ell}\, 3^{\ell} - 4\cdot 2^{k} + 2\cdot 3^{\ell} - 4\cdot 2^{\ell} + 2$$

translate back

$$= 4\left(\frac{n}{m}\right) m^{\log_2 3} - 4n + 2\cdot m^{\log_2 3} - 4m + 2$$

---

count mults: $M_{m,n} = 3\, M_{m/2,\,n/2}$, $M_{1,n} = n$

unroll: $= 3^2 M_{m/4,\,n/4}$ $C_{m,n} = (A_{m,n}, M_{m,n})$

$$\cdots$$

$$= 3^{\ell}\cdot M_{1,\,n/m}$$

$$= m^{\log_2 3}\cdot\left(\frac{n}{m}\right)$$