Exercises

2.1. Use the divide-and-conquer integer multiplication algorithm to multiply the two binary integers 10011011 and 10111010.

2.2. Show that for any positive integers n and any base b, there must some power of b lying in the range $[n, bn]$.

2.3. Section 2.2 describes a method for solving recurrence relations which is based on analyzing the recursion tree and deriving a formula for the work done at each level. Another (closely related) method is to expand out the recurrence a few times, until a pattern emerges. For instance, let's start with the familiar $T(n) = 2T(n/2) + O(n)$. Think of $O(n)$ as being $\leq cn$ for some constant c, so: $T(n) \leq 2T(n/2) + cn$. By repeatedly applying this rule, we can bound $T(n)$ in terms of $T(n/2)$, then $T(n/4)$, then $T(n/8)$, and so on, at each step getting closer to the value of $T(\cdot)$ we do know, namely $T(1) = O(1)$.

$$\begin{aligned} T(n) &\leq 2T(n/2) + cn \\ &\leq 2[2T(n/4) + cn/2] + cn \;=\; 4T(n/4) + 2cn \\ &\leq 4[2T(n/8) + cn/4] + 2cn \;=\; 8T(n/8) + 3cn \\ &\leq 8[2T(n/16) + cn/8] + 3cn \;=\; 16T(n/16) + 4cn \\ &\;\;\vdots \end{aligned}$$

A pattern is emerging... the general term is

$$T(n) \leq 2^k T(n/2^k) + kcn.$$

Plugging in $k = \log_2 n$, we get $T(n) \leq nT(1) + cn \log_2 n = O(n \log n)$.

(a) Do the same thing for the recurrence $T(n) = 3T(n/2) + O(n)$. What is the general kth term in this case? And what value of k should be plugged in to get the answer?

(b) Now try the recurrence $T(n) = T(n-1) + O(1)$, a case which is not covered by the master theorem. Can you solve this too?

Alg A

2.4. Suppose you are choosing between the following three algorithms:

a b d $\Rightarrow$ Caso 3 $\Rightarrow T(n) = O(n^{\log_2 5})$

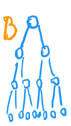
$T(n) = 5T(n/2) + c.n^1$

- Algorithm A solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.

$T(n) = 2T(n-1) + c$

- Algorithm B solves problems of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.

Alg B



$n - h = 1$

$h = n - 1$

$T(n) = O(2^n)$

- Algorithm C solves problems of size n by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

Alg C $T(n) = 9T(n/3) + c.n^2$ Caso 1 $\Rightarrow T(n) = O(n^2 \log n)$

What are the running times of each of these algorithms (in big-O notation), and which would you choose? Alg C, pois $n^2 \log n \in O(n^{2+\epsilon})$, mas a recíproca ñ é verdadeira

2.5. Solve the following recurrence relations and give a Θ bound for each of them.

Caso 3 — (a) $T(n) = 2T(n/3) + 1$

Caso 3 — (b) $T(n) = 5T(n/4) + n$

Caso 1 — (c) $T(n) = 7T(n/7) + n$

Caso 1 — (d) $T(n) = 9T(n/3) + n^2$

$$T(n) = aT(n/b) + c.n^d \begin{cases} 1)\; a = b^d \Rightarrow T(n) = O(n^d \log n) \\ 2)\; a < b^d \Rightarrow T(n) = O(n^d) \\ 3)\; a > b^d \Rightarrow T(n) = O(n^{\log_b a}) \end{cases}$$

2.2) Seja k o maior inteiro tal que $b^k < n$

Temos que $b^{k+1} \geq n \Rightarrow n \leq b^{k+1} = b.b^k < b.n$

ou seja, $b^{k+1} \in [n, bn)$

2.3) a) $T(n) = 3T(n/2) + c.n$

i) Expandindo

$$T(n) = 3T(n/2) + c.n$$
$$T(n/2) = 3T(n/4) + c.n/2$$
$$T(n/4) = 3T(n/8) + c.n/4$$
$$T(n/8) = 3T(n/16) + c.n/8$$
$$\vdots$$

ii) Substituindo

$$T(n) = 3T(n/2) + c.n$$
$$= 3\left(3T(n/4) + c.n/2\right) + c.n$$
$$T(n) = 3^2 T(n/4) + (5/2)c.n$$
$$= 3^2\left(3T(n/8) + c.n/4\right) + (5/2)c.n$$
$$T(n) = 3^3 T(n/8) + (19/4)c.n$$
$$= 3^3\left(3T(n/16) + c.n/8\right) + (19/4).c.n$$
$$= 3^4 T(n/16) + \left(\left(\tfrac{3}{2}\right)^3 + \left(\tfrac{3}{2}\right)^2 + \left(\tfrac{3}{2}\right)^1 + \left(\tfrac{3}{2}\right)^0\right)c.n$$

iii) Generalizando

$$\boxed{T(n) = 3^i T(n/2^i) + \left(\left(\tfrac{3}{2}\right)^i + \left(\tfrac{3}{2}\right)^{i-1} + \dots + \left(\tfrac{3}{2}\right)^0\right).c.n}$$
$$= 3^i T(n/2^i) + \frac{(3/2)^{i+1} - 1}{3/2 - 1}.c.n$$
$$= 3^i T(n/2^i) + \boxed{\left((3/2)^{i+1} - 1\right).2.c.n}$$

iv) Final Recorrência/Caso Base

$$\frac{n}{2^i} = 1 \Rightarrow 2^i = n \Rightarrow \boxed{i = \lg n} \Rightarrow T(n) = 3^{\lg n}.T\left(\frac{n}{2^{\lg n}}\right) + \left(\left(\tfrac{3}{2}\right)^{1+\lg n} - 1\right).2.c.n$$
$$= 3^{\log_3 n / \log_3 2}\, T\left(\frac{n}{n}\right) + \left(\left(\tfrac{3}{2}\right).\frac{3^{\lg n}}{2^{\lg n}} - 1\right).2.c.n$$
$$= n^{\frac{1}{\log_3 2}}.T(1) + \left(\frac{3}{2}.\frac{3^{\lg n}}{n} - 1\right)2.c.n$$
$$= n^{\lg 3}.c + \left(\frac{3}{2}.\frac{n^{\lg 3}}{n} - 1\right).2c.n$$
$$= c\left(n^{\lg 3} + 3n^{\lg 3} - 2n\right) = \Theta\left(n^{\lg 3}\right)$$

2.3) b) $T(n) = T(n-1) + c.n$

(Substituindo) $= \left(T(n-2) + c.(n-1)\right) + c.n = T(n-2) + c(2n-1)$

$= \left(T(n-3) + c.(n-2)\right) + c(2n-1) = T(n-3) + c(3n-3)$

(Generalizando) $T(n) = T(n-i) + c\left(n + (n-1) + (n-2) + \dots + (n-i+1)\right)$

(Caso Base) $n - i = 1 \Rightarrow i = n-1 \Rightarrow T(n) = T(n-(n-1)) + c\left(n + (n-1) + (n-2) + \dots + (n-(n-1)+1)\right)$

$$= T(1) + c.(n+2).(n-1)/2$$
$$= c + c(n+2)(n-1)/2 = O(n^2)$$

Caso 1

(e) $T(n) = 8T(n/2) + n^3$

(f) $T(n) = 49T(n/25) + n^{3/2} \log n$ $> \overset{a}{49}\,T(n/\overset{b}{25}) + n^{\overset{d}{3/2}} \Rightarrow a = 49 < 25^{3/2} = 125 = b^d \Rightarrow$ Caso 2 $\Rightarrow T(n) = \Theta(n^{3/2} \log n)$

(g) $T(n) = T(n-1) + 2$ $= \Theta(n)$

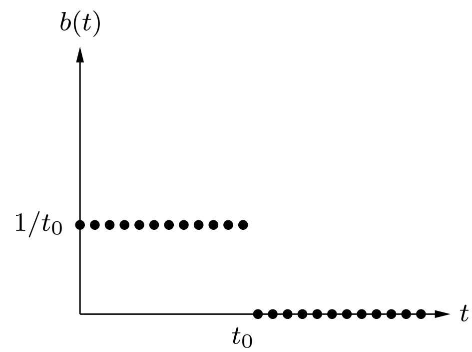
(h) $T(n) = T(n-1) + n^c$, where $c \geq 1$ is a constant $T(n) = \Theta(n^{c+1})$

(i) $T(n) = T(n-1) + c^n$, where $c > 1$ is some constant $T(n) = \sum_{i=1}^{n} c^i = c^n \cdot \sum_{i=1}^{n} \frac{c^i}{c^n} = c^n \sum_{j=0}^{n-1} \frac{1}{c^j} = \Theta(c^n)$

(j) $T(n) = 2T(n-1) + 1$ $= \Theta(2^n)$

(k) $T(n) = T(\sqrt{n}) + 1$ $n = 2^k \Rightarrow T(n) = T(2^k) = T(2^{\frac{k}{2}}) + 1 = \Theta(\lg k) = \Theta(\lg \lg n)$

2.6. A linear, time-invariant system has the following impulse response:



(a) Describe in words the effect of this system.

(b) What is the corresponding polynomial?

2.7. What is the sum of the nth roots of unity? What is their product if n is odd? If n is even?

2.8. Practice with the fast Fourier transform.

(a) What is the FFT of $(1, 0, 0, 0)$? What is the appropriate value of ω in this case? And of which sequence is $(1, 0, 0, 0)$ the FFT?

(b) Repeat for $(1, 0, 1, -1)$.

2.9. Practice with polynomial multiplication by FFT.

(a) Suppose that you want to multiply the two polynomials $x + 1$ and $x^2 + 1$ using the FFT. Choose an appropriate power of two, find the FFT of the two sequences, multiply the results componentwise, and compute the inverse FFT to get the final result.

(b) Repeat for the pair of polynomials $1 + x + 2x^2$ and $2 + 3x$.

2.10. Find the unique polynomial of degree 4 that takes on values $p(1) = 2$, $p(2) = 1$, $p(3) = 0$, $p(4) = 4$, and $p(5) = 0$. Write your answer in the coefficient representation.

2.11. In justifying our matrix multiplication algorithm (Section 2.5), we claimed the following blockwise property: if X and Y are $n \times n$ matrices, and

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}.$$

where A, B, C, D, E, F, G, and H are $n/2 \times n/2$ submatrices, then the product XY can be expressed in terms of these blocks:

$$XY = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

Prove this property.