Universidade Estadual de Campinas – Instituto de Computação

MO417A – Complexidade de Algoritmos – 1º Semestre de 2020

`https://www.ic.unicamp.br/~lehilton/mo417a`

Crescimento de funções

Notação assintótica

Questão 1. (CLRS) Exercícios: 3.1-1, 3.1-2, 3.1-3, 3.1-4, 3.1-5, 3.1-7, 3.2-1, 3.2-2, 3.2-5(*), 3.2-8

Questão 2. (CLRS) Problemas: 3-3

Questão 3. (Dasgupta et al.) Em cada uma das situações abaixo, indique se $f = O(g)$, ou se $f = \Omega(g)$, ou ambos (quando $f = \Theta(g)$). [Justifique formalmente as respostas dos itens (g), (n) e (q). Dica: para (q), suponha que k é constante e compare o quanto cada função cresce para de n para $n+1$.]

	$f(n)$	$g(n)$
(a)	$n - 100$	$n - 200$
(b)	$n^{1/2}$	$n^{2/3}$
(c)	$100n + \log n$	$n + (\log n)^2$
(d)	$n \log n$	$10n \log 10n$
(e)	$\log 2n$	$\log 3n$
(f)	$10 \log n$	$\log(n^2)$
(g)	$n^{1.01}$	$n \log^2 n$
(h)	$n^2/\log n$	$n(\log n)^2$
(i)	$n^{0.1}$	$(\log n)^{10}$
(j)	$(\log n)^{\log n}$	$n/\log n$
(k)	$\sqrt{n}$	$(\log n)^3$
(l)	$n^{1/2}$	$5^{\log_2 n}$
(m)	$n2^n$	3^n
(n)	2^n	2^{n+1}
(o)	$n!$	2^n
(p)	$(\log n)^{\log n}$	$2^{(\log_2 n)^2}$
(q)	$\sum_{i=1}^{n} i^k$	n^{k+1}

Recorrências

Questão 4. (CLRS) Exercícios (3ª edição): 4.3-1, 4.3-2, 4.3-3, 4.3-7, 4.3-9, 4.4-1, 4.4-6, 4.4-8, 4.4-9, 4.5-1, 4.5-4,

Questão 5. (CLRS) 4.3-1 Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

Questão 6. (CLRS) 4.3-2 Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ IS $O(\lg n)$.

Questão 7. (CLRS) 4.3-3 We saw that the solution of $T(n) = T(\lfloor n/2 \rfloor) + n$ is $O(n \lg n)$. Show that the solution of this recurrence is also $\Omega(n \lg n)$. Conclude that the solution is $\Theta(n \lg n)$.

Questão 8. (CLRS) 4.3-7 Using the master method in Section 4.5, you can show that the solution to the recurrence $T(n) = 4T(\lfloor n/3 \rfloor) + n$ is $T(n) = O(n \log_3 4)$. Show that a substitution proof with the assumption $T(n) \leq cn^{\log_3 4}$ fails. Then show how to subtract off a lower-order term to make a substitution proof work.

Questão 9. (CLRS) 4.3-9 Solve the recurrence $T(n) = 3T(\sqrt{n}) + \log n$ by making a change of variables. Your solution should be asymptotically tight. Do not worry about whether values are integral

Questão 10. (CLRS) 4.4-1 Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$. Use the substitution method to verify your answer.

Questão 11. (CLRS) 4.4-6 Argue that the solution to the recurrence $T(n) = T(n/3) + T(2n/3) + cn$, where c is a constant, is $\Omega(n \ln n)$ by appealing to a recursion tree.

Questão 12. (CLRS) 4.4-8 Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(n-a) + T(a) + cn$, where a $a \geq 1$ and $c > 0$ are constants.

Questão 13. (CLRS) 4.4-9 Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.

Questão 14. (CLRS) 4.5-1 Use the master method to give tight asymptotic bounds for the following recurrences.

(a) $T(n) = 2T(n/4) + 1$.
(b) $T(n) = 2T(n/4) + \sqrt{n}$.
(c) $T(n) = 2T(n/4) + n$.
(d) $T(n) = 2T(n/4) + n^2$.

Questão 15. (CLRS) 4.5-4 Can the master method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \lg n$? Why or why not? Give an asymptotic upper bound for this recurrence.

Questão 16. (CLRS) Problemas: 3ª edição: 4-6

Questão 17. (CLRS) Problemas: 2ª edição: 4-7

Questão 18. Encontre a solução da seguinte relação de recorrência. É suficiente encontrar o comportamento assintótico de $T(n)$. Você deve dar argumentos convincentes de que a função $f(n)$ que você encontrou satisfaz $f(n) = \Theta(T(n))$.

$$T(n) = 2T\left(\left\lfloor \frac{n}{\log n} \right\rfloor\right) + 3n, \quad (n > 2), \quad T(1) = 1, \quad T(2) = 2.$$

(Dica: compare essa recorrência com alguma outra recorrência mais fácil de resolver)

Questão 19. Os números de Fibonacci $F(n)$ podem ser estendidos para valores negativos de n usando a mesma definição: $F(n+2) = F(n+1) + F(n)$, e $F(1) = 1$ e $F(0) = 0$. Assim, temos $F(-1) = 1$, $F(-2) = -1$ e assim por diante. Seja $G(n) = F(-n)$. Escreva uma relação de recorrência para $G(n)$ e sugira como resolvê-la. Prove que $G(n) = (-1)^{n+1} F(n)$.