Exercício 3.30. *Prove que o algoritmo mergesort é correto.*

Exercício 3.31. *Faça a análise assintótica do algoritmo merge.*

Exercício 3.32. *Faça a análise assintótica do algoritmo mergesort.*

3.7 O Teorema Mestre

Nesta seção estudaremos as equações de recorrência utilizadas no paradigma de divisão de conquista [4]:

Definição 3.33. *Seja $f(n)$ uma função não-negativa definida no conjunto dos números naturais. Dizemos que $f(n)$ é* eventualmente não-decrescente *se existir um número inteiro n_0 tal que $f(n)$ é não-decrescente no intervalo $[n_0, \infty)$, ou seja,*

$$f(n_1) \leq f(n_2), \forall n_2 > n_1 \geq n_0.$$

Definição 3.34. *Seja $f(n)$ uma função não-negativa definida no conjunto dos números naturais. Dizemos que $f(n)$ é* suave *se for eventualmente não-decrescente e*

$$f(2.n) = \Theta(f(n))$$

Exercício 3.35. *Mostre que $f(n) = n$ é suave.*

Exercício 3.36. *Mostre que $f(n) = \lg n$ é suave.*

Exercício 3.37. *Mostre que $f(n) = n. \lg n$ é suave.*

Exercício 3.38. *Mostre que $f(n) = 3^n$ não é suave.*

Lema 3.39. *Sejam c e n_0 constantes positivas, e $f(n)$ uma função tal que $f(2n) \leq c.f(n), \forall n \geq n_0$. Então $f(2^k n) \leq c^k.f(n), \forall n \geq n_0$ e $k \geq 1$.*

Teorema 3.40. *Seja $f(n)$ uma função suave. Então para qualquer $b \geq 2$ fixado,*

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

O teorema a seguir é conhecido como *regra da suavização*

Teorema 3.41. *Seja $T(n)$ uma função eventualmente não-decrescente, e $f(n)$ uma função suave. Se $T(n) = \Theta(f(n))$ para valores de n que são potências de b ($b \geq 2$), então*

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

A regra da suavização nos permite expandir a informação sobre a ordem de crescimento estabelecida para $T(n)$ de um subconjunto de valores (potências de b) para o domínio inteiro. O teorema a seguir é um resultado muito útil nesta direção conhecido como *teorema mestre*:

Teorema 3.42. *Seja $T(n)$ uma função eventualmente não-decrescente que satisfaz a recorrência*

$T(n) = a.T(n/b) + f(n), \quad$ *para* $n = b^k, k = 1, 2, 3, \ldots$

$T(1) = c$

onde $a \geq 1, b \geq 2$ e $c \geq 0$. Se $f(n) = \Theta(n^d)$, onde $d \geq 0$, então

$$T(n) = \begin{cases} \Theta(n^{\log_b a}), & \text{se } a > b^d \\ \Theta(n^d . \lg n), & \text{se } a = b^d \\ \Theta(n^d), & \text{se } a < b^d \end{cases}$$

Demonstração. Considere que $f(n) = n^d$. Aplicando o método da substituição para a recorrência do teorema, obtemos:

$$T(b^k) = a^k.[T(1) + \sum_{j=1}^{k} f(b^j)/a^j]$$

Como $a^k = a^{\log_b n} = n^{\log_b a}$, podemos reescrever a equação acima como:

$$T(n) = n^{\log_b a}.[T(1) + \sum_{j=1}^{\log_b n} f(b^j)/a^j]$$

e para $f(n) = n^d$, temos:

$$T(n) = n^{\log_b a}.[T(1) + \sum_{j=1}^{\log_b n} (b^j)^d/a^j] = n^{\log_b a}.[T(1) + \sum_{j=1}^{\log_b n} (b^d/a)^j]$$

A soma acima forma uma série geométrica, e portanto:

$$\sum_{j=1}^{\log_b n} (b^d/a)^j = (b^d/a)\frac{(b^d/a)^{\log_b n} - 1}{(b^d/a) - 1}, \text{ se } b^d \neq a.$$

Quando $b^d \neq a$, temos que $\sum_{j=1}^{\log_b n} (b^d/a)^j = \log_b n$. Agora basta analisarmos cada um dos casos: $a < b^d$, $a > b^d$ e $a = b^d$. □

Apresentaremos agora uma versão um pouco mais geral do teorema mestre[3]. Consideraremos como anteriormente uma recorrência da forma:

$$T(n) = a.T(n/b) + f(n)$$

on $a \geq 1$ e $b > 1$ são constantes, e $f(n)$ é uma função assintoticamente positiva.

Teorema 3.43. *Sejam $a \geq 1$ e $b \geq 2$ constantes, $f(n)$ uma função assintoticamente positiva, e $T(n)$ definida nos inteiros não-negativos pela recorrência $T(n) = a.T(n/b) + f(n)$, onde n/b deve ser interpretado como $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$. Então $T(n)$ tem as seguintes cotas assintóticas:*

1. *Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$;*
2. *Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a}. \lg n)$;*
3. *Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$, e se $a.f(n/b) \leq c.f(n)$ para alguma constante $c < 1$, então para todo n suficientemente grande, temos que $T(n) = \Theta(f(n))$.*

A prova será dividida em três lemas, onde inicialmente consideraremos que n é potência de b.

Lema 3.44. *Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função não-negativa definida para potências de b. Defina $T(n)$ para potências de b pela recorrência:*

$$T(n) = \begin{cases} \Theta(1), & se\ n = 1; \\ a.T(n/b) + f(n), & se\ n = b^i \end{cases}$$

onde i é um inteiro positivo. Então

$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j.f(n/b^j).$$

Demonstração. Analise a árvore de recorrência da equação dada. □

Em termos da árvore de recorrência, os três casos do teorema mestre correspondem aos casos onde o custo total da árvore é:

1. dominado pelo custo das folhas;
2. uniformemente distribuído ao longo da árvore;

3. dominado pelo custo da raiz.

Lema 3.45. *Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função não-negativa definida para potências de b. A função $g(n)$ definida para potências de b por:*

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j.f(n/b^j).$$

tem as seguintes cotas assintóticas para potências de b:

1. *Se $f(n) = O(n^{\log_b a-\epsilon})$ para alguma constante $\epsilon > 0$, então $g(n) = O(n^{\log_b a})$;*
2. *Se $f(n) = \Theta(n^{\log_b a})$, então $g(n) = \Theta(n^{\log_b a}.\lg n)$;*
3. *Se $a.f(n/b) \leq c.f(n)$ para alguma constante $c < 1$ e para todo n suficientemente grande, então $g(n) = \Theta(f(n))$.*

Demonstração. Exercício. □

Exercício 3.46. *Resolva as seguintes relações de recorrência:*

1. $T(1) = 1$, $T(n) = 3T(n/2) + n^2$, $n \geq 2$
2. $T(1) = 1$, $T(n) = 2T(n/2) + n$, $n \geq 2$
3. $T(1) \in \Theta(1)$, $T(n) = 3T(n/3 + 5) + n/2$
4. $T(1) = 1$, $T(n) = 2T(n-1) + 1$, $n \geq 2$
5. $T(1) \in \Theta(1)$, $T(n) = 9T(n/3) + n$
6. $T(1) \in \Theta(1)$, $T(n) = T(2n/3) + 1$
7. $T(1) \in \Theta(1)$, $T(n) = 2T(n/4) + 1$
8. $T(1) \in \Theta(1)$, $T(n) = 2T(n/4) + \sqrt{n}$
9. $T(1) \in \Theta(1)$, $T(n) = 2T(n/4) + \sqrt{n}\lg^2 n$
10. $T(1) \in \Theta(1)$, $T(n) = 2T(n/4) + n$
11. $T(1) \in \Theta(1)$, $T(n) = 2T(n/4) + n^2$
12. $T(1) \in \Theta(1)$, $T(n) = 3T(n/2) + nln(n)$
13. $T(1) \in \Theta(1)$, $T(n) = 3T(n/4) + nln(n)$
14. $T(1) \in \Theta(1)$, $T(n) = 2T(n/2) + nln(n)$
15. $T(1) \in \Theta(1)$, $T(n) = 2T(n/2) + n/ln(n)$
16. $T(1) \in \Theta(1)$, $T(n) = T(n-1) + 1/n$
17. $T(1) \in \Theta(1)$, $T(n) = T(n-1) + ln(n)$
18. $T(1) \in \Theta(1)$, $T(n) = \sqrt{n}T(\sqrt{n}) + n$
19. $T(n) = 8T(n/2) + \Theta(n^2)$
20. $T(n) = 8T(n/2) + \Theta(1)$
21. $T(n) = 7T(n/2) + \Theta(n^2)$

4 Leitura complementar:

- [2] (Capítulos 1 e 2)
- [3] (Capítulos 1 e 2)
- [4] (Capítulos 1 e 3)
- [1] (Capítulos 1 e 2)

Referências

[1] Gilles Brassard and Paul Bratley. *Fundamentals of Algorithmics.* Prentice-Hall, Inc., USA, 1996.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, Cambridge, MA, USA, 4 edition, April 2022.

[4] A. V. Levitin. *Introduction to the Design and Analysis of Algorithms, Third Edition.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2012.