Resolução da Lista de exercícios

Exercícios Propostos

1. Considere o seguinte algoritmo:

```
função Pesquisa(n)
  se n ≤ 1
    então 'inspecione elemento' e termine
    senão
      'inspecione cada um dos n elementos'
      Pesquisa(n/3);
  fim se
fim função
```

Qual a complexidade deste algoritmo? Mostre o desenvolvimento das equações de recorrência.

Podemos atribuir à instrução `se...então...senão...fim se` um custo unitário. Ao algoritmo '`inspecione elemento`' podemos arbitrar um custo constante. Por outro lado, o algoritmo relativo a '`inspecione cada um dos n elementos`' é proporcional à quantidade de entrada n. Como o algoritmo é recursivo, o tempo de entrada n é dado pela seguinte relação de recorrência:

$$T(n) = T\left(\frac{n}{3}\right) + n + 1 \tag{1}$$

$$T(1) = 1. \tag{2}$$

Para resolver essa equação, realizamos a seguinte iteração:

$$T\left(n\right) = T\left(\frac{n}{3}\right) + n + 1 \implies \tag{3}$$

$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{3^2}\right) + \frac{n}{3} + 1 \implies \tag{4}$$

$$T\left(\frac{n}{3^2}\right) = T\left(\frac{n}{3^3}\right) + \frac{n}{3^2} + 1 \implies \dots \tag{5}$$

$$\therefore$$

$$T\left(n\right) = T\left(\frac{n}{3^k}\right) + n + \sum_{j=1}^{k} \frac{1}{3^{j-1}} + k, \tag{6}$$

onde k é o número de iterações. Igualando o argumento da função T, do lado direito, a 1 teremos:

$$\frac{n}{3^k} = 1 \implies k = \log_3 n. \tag{7}$$

Precisamos resolver o somatório da direita, que é o somatório dos k primeiros

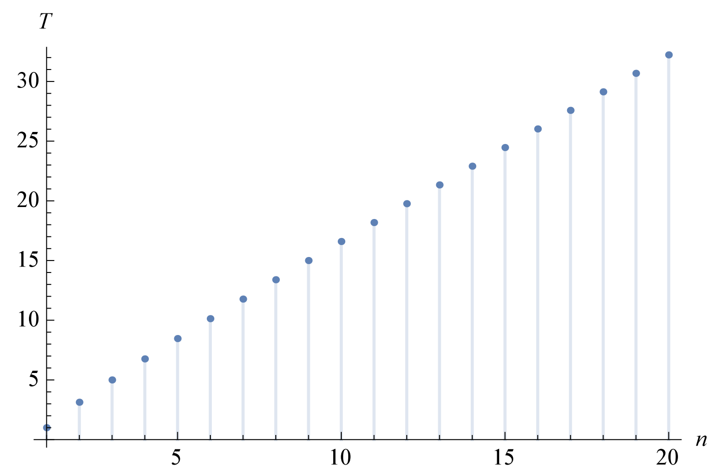

Figura 1: Crescimento do tempo $T(n) = \frac{3n-1+2\log_3 n}{2}$ em função da entrada de tamanho n.

termos da progressão geométrica

$$
\begin{aligned}
S_k &= \frac{a_1\left(q^k-1\right)}{q-1} && (8)\\
&= n\left(1-\frac{1}{3^k}\right)\frac{3}{2} && (9)
\end{aligned}
$$

com razão $q = \frac{1}{3}$. Substituindo em (6)

$$
\begin{aligned}
T(n) &= T(1) + n\left(1-\frac{1}{3^k}\right)\frac{3}{2} + k && (10)\\
&= 1 + n\left(1-\frac{1}{3^{\log_3 n}}\right)\frac{3}{2} + k && (11)\\
&= 1 + n\left(\frac{n-1}{n}\right)\frac{3}{2} + \log_3 n && (12)\\
&= \frac{3n-1+2\log_3 n}{2}. \qquad \square && (13)
\end{aligned}
$$

Considerando a notação *big-O*, a complexidade do algorítimo é $\mathcal{O}(n)$. O crescimento de T em relação à n é ilustrado na Figura 1.

EP-4.2. Resolva as seguintes equações de recorrência:

a) $$\begin{cases} T(n) = T(n-1) + c & , n \geq 2 \\ T(1) = 0; \end{cases}$$

Aplicando a mesma estratégia usada no exercício anterior, temos:

$$T\left(n\right) = T\left(n-k\right) + kc. \tag{14}$$

Fazendo $n - k = 1 \implies k = n - 1$ chegamos em

$$\begin{aligned} T\left(n\right) &= T\left(1\right) + \left(n-1\right)c \qquad (15)\\ &= c\left(n-1\right). \qquad \square \qquad (16) \end{aligned}$$

b) $$\begin{cases} T\left(n\right) = 3T\left(\frac{n}{2}\right) + n & , n \geq 2 \\ T\left(1\right) = 1; \end{cases}$$

As iterações nesta e no próximo item são um pouco mais complicadas, por isso faremos com mais passos. Assim

$$\begin{aligned} T\left(n\right) &= 3T\left(\frac{n}{2}\right) + n \implies \qquad (17)\\ T\left(\frac{n}{2}\right) &= 3T\left(\frac{n}{4}\right) + \frac{n}{2} \implies \qquad (18)\\ T\left(\frac{n}{4}\right) &= 3T\left(\frac{n}{8}\right) + \frac{n}{4} \implies \ldots \qquad (19)\\ &\therefore \\ T\left(n\right) &= 3^k T\left(\frac{n}{2^k}\right) + 3^{k-1}\frac{n}{2^{k-1}} + 3^{k-2}\frac{n}{2^{k-2}} + \ldots + n \qquad (20)\\ &= 3^k T\left(\frac{n}{2^k}\right) + n\sum_{j=0}^{k-1}\frac{3^j}{2^j}. \qquad (21) \end{aligned}$$

Precisamos, primeiramente, resolver o somatório parcial

$$n\sum_{j=0}^{k-1}\frac{3^j}{2^j}, \tag{22}$$

que é o somatório dos k primeiros termos da progressão geométrica

$$\begin{aligned} S_k &= \frac{a_1\left(q^k - 1\right)}{q-1} \implies \qquad (23)\\ n\sum_{j=0}^{k-1}\frac{3^j}{2^j} &= \left(\frac{3^k}{2^k} - 1\right)2n \qquad (24) \end{aligned}$$

com razão $q = \frac{3}{2}$. Assim, tomando a Eq. (21), fazendo $\frac{n}{2^k} = 1$ e as devidas substituições:

$$\begin{aligned} T\left(n\right) &= 3^k T\left(1\right) + 2n\left(\frac{3^k}{2^k} - 1\right) \qquad (25)\\ &= 3^k + \left(\frac{2n3^k}{n} - 2n\right) \qquad (26)\\ &= 3^k + 2\cdot 3^k - 2n \qquad (27)\\ &= 3^{1+\log_2 n} - 2n. \qquad \square \qquad (28) \end{aligned}$$

c) $\begin{cases} T(n) = T\left(\frac{n}{2}\right) + n^2 & , n \geq 2 \\ T(1) = 0. \end{cases}$

Iterando as equações de recorrência

$$T(n) = T\left(\frac{n}{2}\right) + n^2 \implies \tag{29}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^2 \implies \tag{30}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2 \implies \ldots \tag{31}$$

$$\therefore$$

$$T(n) = T\left(\frac{n}{2^k}\right) + \frac{n^2}{2^{(k-1)2}} + \frac{n^2}{2^{(k-2)2}} + \ldots + n^2 \tag{32}$$

$$= T\left(\frac{n}{2^k}\right) + n^2 \sum_{j=0}^{k-1} \frac{1}{4^j}, \tag{33}$$

e tomando a relação em (23) chegamos a

$$T(n) = T\left(\frac{n}{2^k}\right) + n^2 \frac{4}{3}\left(1 - 4^{-k}\right). \tag{34}$$

Faça $n = 2^k \implies \frac{n}{2^k} = 1$ ou $k = \log_2 n$. Assim,

$$T(n) = T(1) + n^2 \frac{4}{3}\left(1 - 4^{-\log_2 n}\right) \tag{35}$$

$$= n^2 \frac{4}{3}\left(1 - 4^{-\log_2 n}\right). \tag{36}$$

É possível simplificar essa última equação para

$$T(n) = \frac{4^{k+1}}{3}\left(1 - 4^{-k}\right). \tag{37}$$

$$= \frac{4^{k+1} - 4^{k+1}4^{-k}}{3} \tag{38}$$

$$= \frac{4^{k+1} - 4}{3} \tag{39}$$

$$= \frac{4}{3}\left(4^{\log_2 n} - 1\right). \qquad \square \tag{40}$$