Projeto e Análise de Algoritmos
2024.2

Algoritmos Recursivos e Relações de Recorrência

Prof. Marcio Costa Santos DCC/UFMG

ALGORITMOS RECURSIVOS

Algoritmo Recursivo 1

[H] inteiro x. ? $x \leq 1$ E $REC(x-1) \geq 1$
$REC(x-1)+1$ 1 REC()

Algoritmo Recursivo 1 - Complexidade

$$F(n) = \begin{cases} 2F(n-1) + 3 & \text{, se } \; n \geq 1 \\ 3 & \text{, em caso contrário} \end{cases}$$

Algoritmo Recursivo 2

[H] raiz de uma árvore r, um inteiro x. ? $r.esq \neq \lambda$
$x \leftarrow x + 1$ REC($r.esq$) x REC()

Algoritmo Recursivo 2 - Complexidade

$$F(r) = \begin{cases} F(r.esq) + 3 & \text{, se } \ r.esq \neq \lambda \\ 2 & \text{, em caso contrário} \end{cases}$$

Algoritmo Recursivo 3

[H] raiz de uma árvore r. ? $r.esq \neq \lambda$
$x \leftarrow REC(r.esq) + 1$ $x \leftarrow 0$ $r.dir \neq \lambda$
$y \leftarrow REC(r.dir) + 1$ $y \leftarrow 0$ maior entre y e x REC()

Algoritmo Recursivo 3 - Complexidade

$$F(r) = \begin{cases} F(r.esq) + F(r.dir) + 7 & \text{, se } \; r.esq \neq \lambda \; \text{ e } \; r.di \\ F(r.esq) + 5 & \text{, se } \; r.esq \neq \lambda \; \text{ e } \; r.di \\ F(r.dir) + 5 & \text{, se } \; r.esq = \lambda \; \text{ e } \; r.di \\ 5 & \text{, em caso contrári} \end{cases}$$

Algoritmo Recursivo 4

[H] um inteiro x. ? $x \geq 1$ $s \leftarrow 0$ $s \leftarrow REC(x-1)$ i $=$ 1 até x $s \leftarrow s + i$ s REC()

Algoritmo Recursivo 4 - Complexidade

$$F(n) = \begin{cases} F(n-1) + 4n + 2 & \text{, se } n \geq 1 \\ 1 & \text{, em caso contrário} \end{cases}$$

Algoritmo Recursivo 5

[H] um inteiro x. ? $x \geq 1$ $s \leftarrow 0$ $s \leftarrow REC(x-1)$
$s \leftarrow s + i$ s REC()

Algoritmo Recursivo 5 - Complexidade

$$F(n) = \begin{cases} F(n-1) + 3 & \text{, se } n \geq 1 \\ 2 & \text{, em caso contrário} \end{cases}$$

RELAÇÕES DE RECORRÊNCIA

Resolução de Recorrências

- Podemos resolver equações na forma

$$T(n) = aT(\frac{n}{b}) + f(n)$$

- Podemos resolver equações na forma

$$T(n) = aT(n-b) + f(n)$$

- Não podemos resolver equações na forma

$$T(n) = T(n-a) + T(n-b)$$

Método da Substituição

- Usar a equação para obter uma fórmula.
- Substituir a expressão em uma versão menor.
- Continuar o processo até o caso base.

Exemplo 1

$$F(n) = \begin{cases} F(n-1) + 3 & \text{, se } \; n \geq 1 \\ 2 & \text{, em caso contrário} \end{cases}$$

Complexidade: $\Theta(n)$

Exemplo 1

$$
\begin{aligned}
F(n) &= F(n-1) + 3 \\
&= (F(n-2) + 3) + 3 \\
&= ((F(n-3) + 3) + 3) + 3 \\
&= \ldots
\end{aligned}
$$

Exemplo 1

$$
\begin{aligned}
F(n) &= F(n-i) + \sum_{j=0}^{i} 3 \\
&= F(0) + \sum_{j=0}^{n} 3 \\
&= 2 + \sum_{j=0}^{n} 3 \\
&= 3n + 2
\end{aligned}
$$

Exemplo 2

$$F(n) = \begin{cases} F(n-1) + 4n + 2 & \text{, se } \; n \geq 1 \\ 1 & \text{, em caso contrário} \end{cases}$$

Complexidade: $\Theta(n^2)$

Exemplo 2

$$
\begin{aligned}
F(n) &= F(n-1) + 4n \\
&= (F(n-2) + 4(n-1) + 2) + 4n \\
&= ((F(n-3) + 4(n-2) + 2) + 4(n-1) + 2) + 4n \\
&=
\end{aligned}
$$

Exemplo 2

$$
\begin{aligned}
F(n) = &\quad F(n-i) + \sum_{j=0}^{i-1} 4(n-j) + \sum_{j=0}^{i-1} 2 \\
= &\quad F(0) + \sum_{j=0}^{n-1} 4(n-j) + \sum_{j=0}^{n-1} 2 \\
= &\quad 2 + n^2 + \frac{n(n-1)}{2} + 2n = \frac{3n^2}{2} + \frac{3n}{2} + 2
\end{aligned}
$$

Exemplo 3

$$F(n) = \begin{cases} 2F(n-1) + 2 & \text{, se } \; n \geq 1 \\ 2 & \text{, em caso contrário} \end{cases}$$

Complexidade: $\Theta(2^n)$

Exemplo 3

$$\begin{aligned} F(n) &= 2F(n-1)+3 \\ &= 2(2F(n-2)+3)+3 \\ &= 2(2(2F(n-3)+3)+3)+3 \\ &= \ldots \end{aligned}$$

Exemplo 3

$$
\begin{aligned}
F(n) &= 2^i F(n-i) + \sum_{j=0}^{i-1} 2^j 3 \\
&= 2^n F(0) + \sum_{j=0}^{n-1} 2^j 3 \\
&= 2^n 2 + \sum_{j=0}^{n-1} 2^j 3 \\
&= 5.2^n - 1
\end{aligned}
$$

Forma Geral

$$F(n) = \begin{cases} aF(n-b) + f(n) & \text{, se } \ n \geq 1 \\ f(0) & \text{, em caso contrário} \end{cases}$$

Complexidade: ?

Forma Geral

$$
\begin{aligned}
F(n) &= aF(n-b) + f(n) \\
&= a(aF(n-2b) + f(n-b)) + f(n) \\
&= a(a(aF(n-3b) + f(n-2b)) + f(n-b)) + f(n) \\
&= \ldots
\end{aligned}
$$

Forma Geral

$$
\begin{aligned}
F(n) = & \quad a^i F(n-ib) + \sum_{j=0}^{i-1} a^j a^j . f(n-jb) \\
= & \quad a^{\frac{n}{b}} F(0) + \sum_{j=0}^{\frac{n}{b}-1} a^j . f(n-jb) \\
= & \quad \Theta((a^{\frac{n}{b}} F(0) + \sum_{j=0}^{\frac{n}{b}-1} a^j . f(n-jb))
\end{aligned}
$$

Exemplo 4

$$F(n) = \begin{cases} F(\frac{n}{2}) + 1 & \text{, se } \ n > 1 \\ 1 & \text{, em caso contrário} \end{cases}$$

Complexidade: $\log_2 n$

Exemplo 4

$$
\begin{aligned}
F(n) &= F(\frac{n}{2}) + 1 \\
&= (F(\frac{n}{2^2}) + 1) + 1 \\
&= ((F(\frac{n}{2^3}) + 1) + 1) + 1 \\
&= \ldots
\end{aligned}
$$

Exemplo 4

$$
\begin{aligned}
F(n) &= F(\frac{n}{2^i}) + \sum_{j=1}^{i} 1 \\
&= F(1) + \sum_{j=1}^{\log_2 n - 1} 1 \\
&= 1 + \log_2 n
\end{aligned}
$$

Exemplo 5

$$F(n) = \begin{cases} 2F(\frac{n}{2}) + 1 & \text{, se } \ n > 1 \\ 1 & \text{, em caso contrário} \end{cases}$$

Complexidade: $\Theta(n)$

Exemplo 5

$$
\begin{aligned}
F(n) &= 2F(\frac{n}{2}) + 1 \\
&= 2(2F(\frac{n}{2^2}) + 1) + 1 \\
&= 2(2(2F(\frac{n}{2^3}) + 1) + 1) + 1 \\
&= \ldots
\end{aligned}
$$

Exemplo 5

$$
\begin{aligned}
F(n) &= 2^i F(\frac{n}{2^i}) + \sum_{j=0}^{i-1} 2^i \\
&= 2^{\log_2 n} F(1) + \sum_{j=0}^{\log_2 n - 1} 2^i \\
&= 2^{\log_2 n} + 2^{\log_2 n} - 1 = 2.2^{\log_2 n} - 1 = 2n - 1
\end{aligned}
$$

Exemplo 6

$$F(n) = \begin{cases} 2F(\frac{n}{3}) + 1 & \text{, se } \; n > 1 \\ 1 & \text{, em caso contrário} \end{cases}$$

Complexidade: $\sqrt[\log_3 2]{n}$

Exemplo 6

$$
\begin{aligned}
F(n) &= 2F(\frac{n}{3}) + 1 \\
&= 2(2F(\frac{n}{3^2}) + 1) + 1 \\
&= 2(2(2F(\frac{n}{3^3}) + 1) + 1) + 1 \\
&= \ldots
\end{aligned}
$$

- 1

Teorema Mestre

Teorema Mestre

Sejam $a \leq 1$ e $b > 1$ constantes, $f(n)$ uma função e

$$T(n) = aT(\frac{n}{b}) + f(n),$$

Então, para algum $\epsilon > 0$

Se $f(n) = O(n^{\log_b a - \epsilon})$ $T(n) = \Theta(n^{\log_b a})$

Se $f(n) = \Theta(n^{\log_b a})$ $T(n) = \Theta(n^{\log_b a} \log n)$

Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$ então
$T(n) = \Theta(f(n))$

Teorema Mestre - Exemplos

- $F(n) = F(\frac{n}{2}) + 1$
- $a = 1$, $b = 2$ e $\log_2 1$
- $\log_2 1 = 0$ então $n^0 = 1$
- $\Theta(n^{\log_b a)} = \theta(1)$ e $f(n) = 1$
- Caso 2! Logo complexidade é $\Theta(n^{\log_b a} \log n) = \Theta(\log n)$

Teorema Mestre - Exemplos

- $F(n) = F(\frac{n}{2}) + n$
- $a = 1$, $b = 2$ e $\log_2 1$
- $\log_2 1 = 0$ então $n^0 = 1$
- $\Theta(n^{\log_b a}) = \Theta(1)$ e $f(n) = n$
- Caso 3! Logo complexidade é $\Theta(n)$

Teorema Mestre - Exemplos

- $F(n) = 2F(\frac{n}{2}) + 1$
- $a = 2$, $b = 2$ e $\log_2 2$
- $\log_2 2 = 1$ então $n^1 = n$
- $\Theta(n^{\log_b a)} = \Theta(n)$ e $f(n) = 1$
- Caso 1! Logo complexidade é $\Theta(n^{\log_b a}) = \Theta(n)$

Teorema Mestre - Exemplos

- $F(n) = 2F(\frac{n}{2}) + n \log n$
- $a = 2,\ b = 2$ e $\log_2 2$
- $\log_2 2 = 1$ então $n^1 = n$
- $\Theta(n^{\log_b a}) = \Theta(n)$ e $f(n) = n \log n$
- Caso 3?
- $n \log n$ não é POLIMONIALMENTE MAIOR QUE n!

Teorema Mestre - Exemplos

- $F(n) = 2F(\sqrt{n}) + n + 1$?
- Substituição de variável $m = \log n$, logo $2^m = n$
- $F(n) = F(2^m) = 2F(\sqrt{2^m}) + 2^m + 1$
- $F(2^m) = 2F(2^{\frac{m}{2}}) + 2^m + 1$
- Defina $G(m) = F(2^m)$
- $G(m) = 2G(\frac{m}{2}) + 2^m + 1$

Teorema Mestre - Exemplos

- $G(m) = 2G(\frac{m}{2}) + 2^m + 1$
- $a = 2$, $b = 2$ e $\log_2 2$
- $\log_2 2 = 1$ então $m^1 = m$
- $\Theta(m^{\log_b a)} = \Theta(m)$ e $f(m) = 2^m + 1$
- Caso 3, logo a complexidade é $\Theta(2^m + 1)$
- $F(n) = G(m) = \Theta(2^m + 1) = \Theta(n + 1) = \Theta(n)$