O Método de Akra-Bazzi na Resolução de Equações de Recorrência

Henrique Felipe (Blog Cyberini)
blogcyberini.blogspot.com.br

01 de dezembro de 2015

1 Introdução

O método de Akra-Bazzi é um dos métodos mais poderosos para solucionar relações de recorrência advindas, principalmente, de algoritmos de divisão e conquista. Tal método abrange uma quantidade muito maior de relações de recorrência que o Teorema Mestre.

Enquanto o Teorema Mestre resolve apenas recorrências do tipo

$$T(n) = aT(n/b) + f(n), \tag{1}$$

com certas restrições em $f(n)$, o método de Akra-Bazzi é capaz de resolver relações de recorrência da forma

$$T(n) = \begin{cases} c_0, & n = x_0 \\ \displaystyle\sum_{i=1}^{k} a_i T\left(b_i n\right) + f(n), & n > x_0 \end{cases} \tag{2}$$

onde $a_i > 0$, $0 < b_i < 1$, $c_0 \in \mathbb{R}$, $x_0 \in \mathbb{N}$, $k \geq 1$ e $f(n)$ é uma função que deve satisfazer a condição de crescimento polinomial[1].

Percebe-se que o método de Akra-Bazzi é bem mais flexível. Além disso, ele é capaz de solucionar qualquer relação de recorrência que o Teorema Mestre é capaz de resolver. Outro aspecto importante é que o Teorema Mestre resolve relações de recorrência nas quais os subproblemas possuem

[1] A versão original do método é bem mais detalhada. Aqui, foram feitas algumas simplificações que serão suficientes para os objetivos do presente texto, entretanto a "essência" do método é a mesma.

o mesmo tamanho. Por outro lado, o método de Akra-Bazzi não possui tal restrição.

Neste texto, apresentaremos algumas relações de recorrências clássicas e resolveremos pelo método de Akra-Bazzi, com o intuito de mostrar o seu potencial. Também resolveremos algumas relações que o Teorema Mestre é incapaz de resolver.

2 O método

Conforme exposto, o método de Akra-Bazzi resolve relações de recorrência da forma

$$T(n) = \begin{cases} c_0, & n = x_0 \\ \displaystyle\sum_{i=1}^{k} a_i T(b_i n) + f(n), & n > x_0 \end{cases} \tag{3}$$

onde

1. $a_i > 0$;
2. $0 < b_i < 1$;
3. $f(n)$ é uma função real, positiva e que satisfaz a condição de crescimento polinomial. De forma simplificada, se $|f'(n)|$ tiver como limitante superior algum polinômio, ou seja, $|f'(n)| = O(n^a),\ a \in \mathbb{N}$, então $f(n)$ satisfaz a condição de crescimento polinomial;
4. $k \geq 1$;
5. $c_0 \in \mathbb{R}$.
6. $x_0 \in \mathbb{N}$.

Assim, considerando tal relação de recorrência e, seja a constante p, a única raiz real da equação

$$\sum_{i=1}^{k} a_i b_i^p = 1, \tag{4}$$

então,

$$T(n) = \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right), \tag{5}$$

onde n_0 é uma constante suficientemente grande[2].

Os dois pontos fundamentais da equação são justamente a equação (4) e a integral em (5). Em particular, nem sempre é possível resolver analiticamente a equação (4), o que nos leva a recorrer aos métodos numéricos. Por outro lado, a integral em (5) pode ser complicada de se resolver dependendo de como é a função $f(n)$.

3 Exemplos clássicos

Agora que o método de Akra-Bazzi foi apresentado, vamos utilizá-lo na resolução de algumas relações de recorrência clássicas.

3.1 A busca binária

A relação de recorrência da busca binária recursiva é dada por

$$T(n) = \begin{cases} c_0, & n = 1 \\ T(n/2) + c_1, & n > 1 \end{cases} \tag{6}$$

onde c_0 e c_1 são constantes. Por simplicidade, mas sem perda de generalidade, assumiremos que $c_0 = c_1 = 1$, ou seja,

$$T(n) = \begin{cases} 1, & n = 1 \\ T(n/2) + 1, & n > 1. \end{cases} \tag{7}$$

Pelo método de Akra-Bazzi, temos que $k = 1$ (pois só há um termo envolvendo $T(n)$), $f(n) = 1$, $a_1 = 1$, $b_1 = \frac{1}{2}$. $f(n)$ obviamente satisfaz a condição

[2]Podemos, por exemplo, escolher n_0 de tal forma que o limite inferior da integral seja zero, ou seja, simplesmente o ignoramos, pois contribuirá apenas em um termo constante, que será "absorvido" por termos com maior ordem de grandeza.

de crescimento polinomial, pois $|f'(n)| = 0 = O(1)$. Pela equação (4), temos

$$\begin{aligned}
\sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p = 1 \times \left(\frac{1}{2}\right)^p &= 1 \\
\left(\frac{1}{2}\right)^p &= 1 \\
\log_2 \left(\frac{1}{2}\right)^p &= \log_2 1 \\
p \log_2 \left(\frac{1}{2}\right) &= 0 \\
-p &= 0 \\
p &= 0.
\end{aligned}$$

Portanto, pela equação (5)

$$\begin{aligned}
T(n) &= \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right) \\
T(n) &= \Theta\left(n^0 + n^0 \int_{n_0}^{n} \frac{1}{x^{0+1}}\, dx\right) \\
T(n) &= \Theta\left(1 + 1 \times \int_{n_0}^{n} \frac{1}{x}\, dx\right) \\
T(n) &= \Theta\left(1 + \int_{n_0}^{n} \frac{1}{x}\, dx\right) \\
T(n) &= \Theta\left(1 + \ln x|_{n_0}^{n}\right) \\
T(n) &= \Theta\left(1 + \ln n - \ln n_0\right)
\end{aligned}$$

escolhendo $n_0 = 1$, temos

$$T(n) = \Theta\left(1 + \ln n\right) = \Theta\left(\ln n\right),$$

que é um resultado já conhecido. Não devemos nos preocupar com a base do logaritmo, pois isso influencia apenas por fatores constantes.

3.2 *MergeSort*

De maneira simplificada, mas sem perda de generalidade, a relação de recorrência do algoritmo de ordenação por intercalação (*MergeSort*) é dada por

$$T(n) = \begin{cases} 1, & n = 1 \\ 2T(n/2) + n, & n > 1. \end{cases} \tag{8}$$

Novamente, igualamos algumas das constantes a um, pois influenciariam apenas por fatores constantes.

Pelo método de Akra-Bazzi, temos que $k = 1$ (pois só há um termo envolvendo $T(n)$), $f(n) = n$, $a_1 = 2$, $b_1 = \frac{1}{2}$. $f(n)$ obviamente satisfaz a condição de crescimento polinomial, pois $|f'(n)| = 1 = O(1)$. Pela equação (4), temos

$$\begin{aligned}
\sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p = 2 \times \left(\frac{1}{2}\right)^p &= 1 \\
\left(\frac{1}{2}\right)^p &= \frac{1}{2} \\
\log_2 \left(\frac{1}{2}\right)^p &= \log_2 \frac{1}{2} \\
p \log_2 \left(\frac{1}{2}\right) &= -1 \\
-p &= -1 \\
p &= 1
\end{aligned}$$

Portanto, pela equação (5)

$$\begin{aligned}
T(n) &= \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right) \\
T(n) &= \Theta\left(n^1 + n^1 \int_{n_0}^{n} \frac{x}{x^{1+1}}\, dx\right) \\
T(n) &= \Theta\left(n + n \int_{n_0}^{n} \frac{x}{x^2}\, dx\right) \\
T(n) &= \Theta\left(n + n \int_{n_0}^{n} \frac{1}{x}\, dx\right) \\
T(n) &= \Theta\left(n + n \left(\ln x|_{n_0}^{n}\right)\right) \\
T(n) &= \Theta\left(n + n \ln n - n \ln n_0\right)
\end{aligned}$$

escolhendo $n_0 = 1$, temos

$$T(n) = \Theta\left(n + n \ln n\right) = \Theta\left(n \ln n\right),$$

que é um resultado já conhecido. Já que $n \ln n$ é assintoticamente maior que n, então o termo n foi descartado. Novamente, não devemos nos preocupar com a base do logaritmo, pois isso influencia apenas por fatores constantes.

Repare que a mesma equação de recorrência é válida para o melhor caso do *QuickSort*, que ocorre quando o particionamento é balanceado.

3.3 Algoritmo de Strassen

O algoritmo de Strassen realiza multiplicações de matrizes utilizando o paradigma de divisão e conquista. A relação de recorrência de tal algoritmo é dada por

$$T(n) = \begin{cases} 1, & n = 1 \\ 7T(n/2) + n^2, & n > 1. \end{cases} \tag{9}$$

Algumas simplificações foram realizadas, mas o resultado final não será afetado.

Pelo método de Akra-Bazzi, temos que $k = 1$, $f(n) = n^2$, $a_1 = 7$, $b_1 = \frac{1}{2}$. $f(n)$ obviamente satisfaz a condição de crescimento polinomial, pois $|f'(n)| = 2n = O(n)$. Pela equação (4), temos

$$\begin{aligned} \sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p = 7 \times \left(\frac{1}{2}\right)^p &= 1 \\ \left(\frac{1}{2}\right)^p &= \frac{1}{7} \\ \log_2 \left(\frac{1}{2}\right)^p &= \log_2 \frac{1}{7} \\ p \log_2 \left(\frac{1}{2}\right) &= -\log_2 7 \\ -p &= -\log_2 7 \\ p &= \log_2 7 \end{aligned}$$

Portanto, pela equação (5)

$$T(n) = \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + n^{\log_2 7} \int_{n_0}^{n} \frac{x^2}{x^{\log_2 7+1}}\, dx\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + n^{\log_2 7} \int_{n_0}^{n} \frac{x^2}{x \times x^{\log_2 7}}\, dx\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + n^{\log_2 7} \int_{n_0}^{n} \frac{x}{x^{\log_2 7}}\, dx\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + n^{\log_2 7} \int_{n_0}^{n} x^{1-\log_2 7}\, dx\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + n^{\log_2 7} \left(\left.\frac{x^{2-\log_2 7}}{2-\log_2 7}\right|_{n_0}^{n}\right)\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + \frac{n^{\log_2 7}}{2-\log_2 7}\left(n^{2-\log_2 7} - n_0^{2-\log_2 7}\right)\right)$$

$$T(n) = \Theta\left(n^{\log_2 7} + \frac{1}{2-\log_2 7}\left(n^2 - n_0^{2-\log_2 7} n^{\log_2 7}\right)\right)$$

para n_0 suficientemente grande, temos

$$T(n) = \Theta\left(n^{\log_2 7} + \frac{n^2}{2-\log_2 7}\right).$$

Como $\log_2 7 > 2$, então

$$T(n) = \Theta\left(n^{\log_2 7}\right) = \Theta\left(n^{2{,}807\ldots}\right),$$

que é justamente a solução. Observamos que o algoritmo de Strassen é ligeiramente mais eficaz que o algoritmo clássico de multiplicação de matrizes, cuja complexidade é $T(n) = \Theta\left(n^3\right)$.

3.4 *QuickSort* não balanceado

Até agora, todos os exemplos expostos podiam ser resolvidos através do Teorema Mestre. A partir deste exemplo, nenhuma relação de recorrência poderá ser resolvida através dele.

Suponhamos que, ao utilizar o *QuickSort*, a divisão do problema gera dois subproblemas: um cujo tamanho é igual um quinto do original e o outro igual a quatro quintos do original. Se a cada chamada recursiva essa subdivisão se manter, então a relação de recorrência simplificada será

$$T(n) = \begin{cases} 1, \quad n = 1 \\ T(n/5) + T(4n/5) + n, \quad n > 1. \end{cases} \tag{10}$$

Tal relação de recorrência obviamente não pode ser resolvida pelo Teorema Mestre, muito menos pela árvore de recursão. Por outro lado, usar indução também não é uma boa ideia, pois para isso é necessário "chutar" a solução da recorrência e provar o "chute" via indução, ou seja, a indução não resolve a relação de recorrência. Felizmente, o método de Akra-Bazzi é capaz de resolver a recorrência e nos dar um resultado interessante.

Pelo método, temos que $k = 2$ (pois há dois termos envolvendo $T(n)$), $f(n) = n$, $a_1 = 1$, $b_1 = \frac{1}{5}$, $a_2 = 1$, $b_2 = \frac{4}{5}$. $f(n)$ obviamente satisfaz a condição de crescimento polinomial. Pela equação (4), temos

$$\sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p + a_2 b_2^p = 1 \times \left(\frac{1}{5}\right)^p + 1 \times \left(\frac{4}{5}\right)^p = 1$$
$$\left(\frac{1}{5}\right)^p + \left(\frac{4}{5}\right)^p = 1.$$

Essa equação não pode ser resolvida por métodos convencionais. Mas é fácil de ver que se $p = 1$, então ela é automaticamente satisfeita.

Portanto, pela equação (5)

$$T(n) = \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right)$$
$$T(n) = \Theta\left(n^1 + n^1 \int_{n_0}^{n} \frac{x}{x^{1+1}}\, dx\right)$$
$$T(n) = \Theta\left(n + n \int_{n_0}^{n} \frac{x}{x^2}\, dx\right),$$

que é igual exemplo do *MergeSort*. Portanto,

$$T(n) = \Theta\left(n \ln n\right). \tag{11}$$

Ou seja, mesmo quando a divisão não é balanceada, ou seja, os subproblemas têm tamanhos desiguais, o *QuickSort* ainda tem complexidade $\Theta\left(n \ln n\right)$. De forma geral, quando o particionamento não é balanceado, temos

$$T(n) = \begin{cases} 1, \quad n = 1 \\ T(b_1 n) + T(b_2 n) + n, \quad n > 1 \end{cases} \tag{12}$$

onde $b_1, b_2 \in (0,1)$ e $b_1 + b_2 = 1$. Pela equação (4), temos que p sempre será 1, independente dos valores de b_1 e b_2, pois teremos

$$b_1^p + b_2^p = 1 \Leftrightarrow b_1 + b_2 = 1, \text{ quando } p = 1. \tag{13}$$

Portanto, $T(n)$ sempre será $\Theta(n \ln n)$.

4 Exemplos gerais

Vamos expor agora mais alguns exemplos de relações de recorrência que não podem ser resolvidas pelo Teorema Mestre.

4.1 Exemplo 1

Seja[3],

$$T(n) = \begin{cases} 1, & n = 1 \\ 3T(n/3) + n \ln n, & n > 1. \end{cases} \tag{14}$$

Pelo método de Akra-Bazzi, temos que $k = 1$, $f(n) = n \ln n$, $a_1 = 3$, $b_1 = \dfrac{1}{3}$. $f(n)$ obviamente satisfaz a condição de crescimento polinomial, pois $|f'(n)| = |\ln n + 1| = O(n)$. Pela equação (4), temos

$$\begin{aligned} \sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p = 3 \times \left(\frac{1}{3}\right)^p &= 1 \\ \left(\frac{1}{3}\right)^p &= \frac{1}{3} \\ p &= 1 \end{aligned}$$

Portanto, pela equação (5)

$$\begin{aligned} T(n) &= \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right) \\ T(n) &= \Theta\left(n^1 + n^1 \int_{n_0}^{n} \frac{x \ln x}{x^{1+1}}\, dx\right) \\ T(n) &= \Theta\left(n + n \int_{n_0}^{n} \frac{x \ln x}{x^2}\, dx\right) \\ T(n) &= \Theta\left(n + n \int_{n_0}^{n} \frac{\ln x}{x}\, dx\right). \end{aligned}$$

[3]A escolha de $\ln n$ no lugar de $\log n$ não influencia no resultado final, já que diferem apenas por um fator constante. O uso de $\ln n$ tem como objetivo facilitar o cálculo das integrais.

Pelo método da substituição das integrais, temos que

$$u = \ln x, \quad du = \frac{1}{x}dx,$$
$$u(n) = \ln n, \quad u(n_0) = \ln n_0.$$

Logo,

$$T(n) = \Theta\left(n + n\int_{\ln n_0}^{\ln n} u\ du\right)$$
$$T(n) = \Theta\left(n + n\left(\frac{u^2}{2}\Big|_{\ln n_0}^{\ln n}\right)\right)$$
$$T(n) = \Theta\left(n + \frac{n}{2}\left(\ln^2 n - \ln^2 n_0\right)\right),$$

escolhendo $n_0 = 1$, temos

$$T(n) = \Theta\left(n + \frac{1}{2}n\ln^2 n\right) = \Theta\left(n\ln^2 n\right). \tag{15}$$

4.2 Exemplo 2

Seja a relação de recorrência

$$T(n) = \begin{cases} 1, \quad n = 1 \\ T(n/2) + T(n/4) + T(n/8) + 1, \quad n > 1. \end{cases} \tag{16}$$

Pelo método de Akra-Bazzi, temos que $k = 3$, $f(n) = 1$, $a_1 = 1$, $b_1 = \frac{1}{2}$, $a_2 = 1$, $b_2 = \frac{1}{4}$, $a_3 = 1$ e $b_3 = \frac{1}{8}$. $f(n)$ obviamente satisfaz a condição de crescimento polinomial, pois $|f'(n)| = 0 = O(1)$. Pela equação (4), temos

$$\sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p + a_2 b_2^p + a_3 b_3^p = 1 \times \left(\frac{1}{2}\right)^p + 1 \times \left(\frac{1}{4}\right)^p + 1 \times \left(\frac{1}{8}\right)^p = 1$$
$$\left(\frac{1}{2}\right)^p + \left(\frac{1}{4}\right)^p + \left(\frac{1}{8}\right)^p = 1.$$

Fazendo a mudança de variável $x = \left(\frac{1}{2}\right)^p$, temos

$$x + x^2 + x^3 = 1$$
$$x^3 + x^2 + x - 1 = 0,$$

que é uma equação completa de terceiro grau, cuja solução exata pode ser obtida pela fórmula de Tartaglia-Cardano. A solução é dada por $x = 0,54368901\ldots$. Logo,

$$\begin{aligned}
\left(\frac{1}{2}\right)^p &= x \\
\log_2 \left(\frac{1}{2}\right)^p &= \log_2 x \\
p \log_2 \left(\frac{1}{2}\right) &= \log_2 x \\
-p &= \log_2 x \\
p &= -\log_2 x \\
p &= -\log_2 0,54368901\ldots \\
p &= 0,8791\ldots
\end{aligned}$$

Por simplicidade, trabalharemos apenas com três casas decimais, ou seja, $p \cong 0,879$. Pela equação (5)

$$\begin{aligned}
T(n) &= \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right) \\
T(n) &= \Theta\left(n^{0,879} + n^{0,879} \int_{n_0}^{n} \frac{1}{x^{0,879+1}}\, dx\right) \\
T(n) &= \Theta\left(n^{0,879} + n^{0,879} \int_{n_0}^{n} \frac{1}{x^{1,879}}\, dx\right) \\
T(n) &= \Theta\left(n^{0,879} + n^{0,879} \int_{n_0}^{n} x^{-1,879}\, dx\right) \\
T(n) &= \Theta\left(n^{0,879} + n^{0,879} \left(\left.\frac{x^{-0,879}}{-0,879}\right|_{n_0}^{n}\right)\right) \\
T(n) &= \Theta\left(n^{0,879} - \frac{n^{0,879}}{0,879}\left(n^{-0,879} - n_0^{-0,879}\right)\right) \\
T(n) &= \Theta\left(n^{0,879} - \frac{1}{0,879}\left(1 - n_0^{-0,879} n^{0,879}\right)\right),
\end{aligned}$$

para n_0 suficientemente grande, temos

$$T(n) = \Theta\left(n^{0,879} - \frac{1}{0,879}\right) = \Theta\left(n^{0,879\ldots}\right). \tag{17}$$

4.3 Exemplo 3

Seja,

$$T(n) = \begin{cases} 15, & n = 1 \\ T(n/3) + T(n/2) + n, & n > 1. \end{cases} \tag{18}$$

Pelo método de Akra-Bazzi, temos que $k = 2$, $f(n) = n$, $a_1 = 1$, $b_1 = \dfrac{1}{3}$, $a_2 = 1$, $b_2 = \dfrac{1}{2}$. $f(n)$ satisfaz a condição de crescimento polinomial, pois $|f'(n)| = 1 = O(1)$. Pela equação (4), temos

$$\begin{aligned} \sum_{i=1}^{k} a_i b_i^p = a_1 b_1^p + a_2 b_2^p = 1 \times \left(\frac{1}{3}\right)^p + 1 \times \left(\frac{1}{2}\right)^p &= 1 \\ \left(\frac{1}{3}\right)^p + \left(\frac{1}{2}\right)^p &= 1. \end{aligned}$$

Resolvendo numericamente, obtemos $p = 0,787\ldots$. Pela equação (5),

$$\begin{aligned} T(n) &= \Theta\left(n^p + n^p \int_{n_0}^{n} \frac{f(x)}{x^{p+1}}\, dx\right) \\ T(n) &= \Theta\left(n^{0,787} + n^{0,787} \int_{n_0}^{n} \frac{x}{x^{0,787+1}}\, dx\right) \\ T(n) &= \Theta\left(n^{0,787} + n^{0,787} \int_{n_0}^{n} \frac{x}{x^{1,787}}\, dx\right) \\ T(n) &= \Theta\left(n^{0,787} + n^{0,787} \int_{n_0}^{n} x^{-0,787}\, dx\right) \\ T(n) &= \Theta\left(n^{0,787} + \frac{n^{0,787}}{1 - 0,787}\left(x^{1-0,787}\Big|_{n_0}^{n}\right)\right) \\ T(n) &= \Theta\left(n^{0,787} + \frac{n^{0,787}}{1 - 0,787}\left(n^{1-0,787} - n_0^{1-0,787}\right)\right) \\ T(n) &= \Theta\left(n^{0,787} + \frac{1}{1 - 0,787}\left(n - n_0^{1-0,787} n^{0,787}\right)\right), \end{aligned}$$

escolhendo $n_0 = 0$, temos

$$T(n) = \Theta\left(n^{0,787} + \frac{n}{1 - 0,787}\right) = \Theta(n). \tag{19}$$

5 Conclusões

Conforme o texto apresenta, o método de Akra-Bazzi certamente é muito poderoso ao resolver relações de recorrência, sendo superior ao Teorema Mestre. As principais "dificuldades" encontradas em seu emprego é justamente a necessidade do cálculo de integrais e encontrar a constante p, utilizada pela fórmula do método.

É claro, tais dificuldades podem ser superadas facilmente, pois, para determinar p, pode-se optar pelo uso de algum método numérico ou de um sistema de computação algébrica. Por outro lado, no cálculo da integral, pode-se também utilizar um sistema de computação algébrica.

O método de Akra-Bazzi também pode ser visto como uma das inúmeras justificativas para o estudo de Cálculo Diferencial e Integral e Cálculo Numérico em cursos superiores de computação. Apesar do método consistir basicamente no emprego de fórmulas matemáticas que podem ser resolvidas facilmente num computador, é importante ter consciência de onde vem os resultados obtidos ao empregar o método.

Por fim, é importante lembrar que tal método não é capaz de resolver recorrências do tipo

$$T(n) = \sum_{i=0}^{k} a_i T(n - b_i) + f(n),$$

que surgem, por exemplo, ao analisar a versão recursiva do algoritmo da sequência de Fibonacci ou a versão recursiva do fatorial. Para tais casos, outros método devem ser empregados.

Referências

[1] CORMEN, T. H. et al. *Algoritmos: teoria e prática.* 3 ed. Rio de Janeiro: Elsevier, 2012.

[2] Akra, M., Bazzi, L. (1998). On the solution of linear recurrence equations. *Computational Optimization and Applications, 10(2)*, 195-210.