MC358 - Fundamentos matemáticos da computação

Prof. Dr. Hilder Vitor Lima Pereira

Instituto de computação

UNICAMP

08 de novembro de 2023

1 Recorrências

2 Perguntas, observações, comentários?

Recorrências

"Para saber o valor da sequência, primeiro você precisa saber o valor da sequência."

Sequências ou funções que envolvem recorrências, são definidas com respeito a si próprias, mas incluem um caso base:

- $a_0 = 3$, $a_n = 2a_{n-1} + 5$ para $n \geq 1$.
- $f(n) = \begin{cases} 3 & \text{se } n = 0 \\ 2f(n-1) + 5 & \text{se } n \geq 1 \end{cases}$
- $b_0 = b_1 = 100$, $b_n = b_{n-1} + 5b_{n-2} + 2$ para $n \geq 1$.
- $g(n) = \begin{cases} 3 & \text{se } n \leq 1 \\ 2g(n/2) + 5 & \text{se } n > 1 \end{cases}$

Calcular $f(k)$ para k pequeno envolve apenas alguns níveis recursivos.

Por exemplo, considere: $f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$

Calcule $f(3)$ (exemplo na lousa).

Calcular $f(k)$ para k pequeno envolve apenas alguns níveis recursivos.

Por exemplo, considere: $f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$

Calcule $f(3)$ (exemplo na lousa).

Agora calcule $f(1000)$.

Calcular $f(k)$ para k pequeno envolve apenas alguns níveis recursivos.

Por exemplo, considere: $f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1)+1 & \text{se } n \geq 1 \end{cases}$

Calcule $f(3)$ (exemplo na lousa).

Agora calcule $f(1000)$.

Nem o Python consegue... Mesmo assim, sei que $f(1000)$ é

3214525821558801962845275147180005431684214435116600822331251165111053153374808367479595136447
0875743827840187526594404755614358570769421307953732724095724411803703324472692956263223815187
1134256338625464591394249507458238021963026774966318382311887437135894330596265029812894949578
7316051161700420812 7

Como calculei $f(1000)$?

Fórmula fechada para recorrências

Muitas vezes, é possível achar uma fórmula não recursiva para uma função que envolve recorrências (neste caso, dizemos que resolvemos a ou achamos uma solução para a recorrência).

Fórmula fechada para recorrências

Muitas vezes, é possível achar uma fórmula não recursiva para uma função que envolve recorrências (neste caso, dizemos que resolvemos a ou achamos uma solução para a recorrência).

No exemplo anterior,

$$f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$$

é possível mostrar que

$$f(n) = 3 \cdot 2^n - 1.$$

Resolvendo recorrências

- Iteração e substituição
- Equação característica
- Estimativa assintótica

Vamos começar com a função

$$f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$$

Aplicamos a definição da função algumas vezes para tentar obter um candidato, um palpite, para o termo geral.

Recorrências com apenas um termo recursivo

Vamos começar com a função

$$f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$$

Aplicamos a definição da função algumas vezes para tentar obter um candidato, um palpite, para o termo geral.
Temos

$$f(n) = 2^k \cdot f(n-k) + 2^k - 1$$

Recorrências com apenas um termo recursivo

Vamos começar com a função

$$f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$$

Aplicamos a definição da função algumas vezes para tentar obter um candidato, um palpite, para o termo geral.
Temos

$$f(n) = 2^k \cdot f(n-k) + 2^k - 1$$

Usamos o caso base para obter uma expressão para $f(n)$:

$$n = k \Rightarrow f(n) = 2^n f(0) + 2^n - 1 = 3 \cdot 2^n - 1$$

Recorrências com apenas um termo recursivo

Vamos começar com a função

$$f(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2f(n-1) + 1 & \text{se } n \geq 1 \end{cases}$$

Aplicamos a definição da função algumas vezes para tentar obter um candidato, um palpite, para o termo geral.
Temos

$$f(n) = 2^k \cdot f(n-k) + 2^k - 1$$

Usamos o caso base para obter uma expressão para $f(n)$:

$$n = k \Rightarrow f(n) = 2^n f(0) + 2^n - 1 = 3 \cdot 2^n - 1$$

Usamos indução matemática (ou PBO) para provar que $f(n)$ realmente tem a fórmula fechada que encontramos.

Exemplo:

$$f(n) = \begin{cases} 5 & \text{se } n \in \{0, 1\} \\ 3f(n-2) + 2 & \text{se } n \geq 2 \end{cases}$$

Exemplo com algoritmo

Algorithm: A

Input: $\{v_1, ..., v_n\} \subset \mathbb{Z}$.

Output: Um valor inteiro

if $1 == n$ then
    return $\max(v_1, 0)$
$v_1 = \mathtt{A}(\{v_1, ..., v_{n-1}\})$
$v_n = \mathtt{A}(\{v_1, ..., v_{n-1}\})$
for $1 \leq i \leq n$ do
    if $0 == v_1$ then
        $v_i = 10 - v_1$
return $v_1 + v_n$

Defina $f(n)$ como a quantidade de *ifs* que o algoritmo executa em uma entrada de tamanho n.
Sabendo que $\sum_{i=1}^{m} i \cdot 2^i = (m-1) \cdot 2^{m+1} + 2$, ache uma fórmula fechada para $f(n)$.

Perguntas, observações, comentários?