Resolução de Recorrências: Método da Substituição

Prof. Daniel Kikuti

Universidade Estadual de Maringá

Introdução

Definição

Uma **recorrência** é uma função definida em termos de:

- um ou mais casos base;
- de si mesma ("valores anteriores" da mesma função).

O que significa resolver uma recorrência?

Resolver uma recorrência é encontrar uma fórmula fechada (expressão equivalente) que dê o valor da função diretamente em termos do seu argumento (sem recursão).

Exemplo

A somatória $\sum_{i=1}^{n} i$ pode ser definida pela recorrência:

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

A fórmula fechada para esta recorrência é:

$$f(n) = \frac{n(n+1)}{2}$$

Aquecimento

O problema das Torres de Hanoi

Dados três pinos e 8 discos inicialmente empilhados em ordem decrescente de tamanho em um dos pinos, transfira a torre inteira para um dos outros pinos, movendo um disco de cada vez e nunca colocando um disco maior em cima de um menor.

Quantos movimentos são necessários e suficientes?

Seja $T(n)$ o número mínimo de movimentos para transferir n discos de um pino para o outro segundo as regras.

Aquecimento

O problema das Torres de Hanoi

Dados três pinos e 8 discos inicialmente empilhados em ordem decrescente de tamanho em um dos pinos, transfira a torre inteira para um dos outros pinos, movendo um disco de cada vez e nunca colocando um disco maior em cima de um menor.

Quantos movimentos são necessários e suficientes?

Seja $T(n)$ o número mínimo de movimentos para transferir n discos de um pino para o outro segundo as regras.

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

A recorrência permite computar $T(n)$ para qualquer n que quisermos.

Aquecimento

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

Então como resolver a recorrência?

Um jeito é **adivinhar** a solução correta e provar que seu chute estava correto.

Aquecimento

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

Então como resolver a recorrência?

Um jeito é **adivinhar** a solução correta e provar que seu chute estava correto.

Como é proibido usar bola de cristal na prova...

Considere os seguintes valores calculados:

$$\begin{array}{rclclcr} T(2) & = & 2T(1)+1 & = & 2.1+1 & = & 3 \\ T(3) & = & 2T(2)+1 & = & 2.3+1 & = & 7 \\ T(4) & = & 2T(3)+1 & = & 2.7+1 & = & 15 \\ T(5) & = & 2T(4)+1 & = & 2.15+1 & = & 31 \\ T(6) & = & 2T(5)+1 & = & 2.31+1 & = & 63 \end{array}$$

Algum padrão?

Aquecimento

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

Então como resolver a recorrência?

Um jeito é **adivinhar** a solução correta e provar que seu chute estava correto.

Como é proibido usar bola de cristal na prova...

Considere os seguintes valores calculados:

$$\begin{array}{rcrcrcr} T(2) & = & 2T(1)+1 & = & 2.1+1 & = & 3 \\ T(3) & = & 2T(2)+1 & = & 2.3+1 & = & 7 \\ T(4) & = & 2T(3)+1 & = & 2.7+1 & = & 15 \\ T(5) & = & 2T(4)+1 & = & 2.15+1 & = & 31 \\ T(6) & = & 2T(5)+1 & = & 2.31+1 & = & 63 \end{array}$$

Algum padrão? Certamente $T(n) = 2^n - 1$.

Aquecimento

Mostraremos por **indução matemática** que $T(n) = 2^n - 1$ é a fórmula fechada para a recorrência:

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

Aquecimento

Mostraremos por **indução matemática** que $T(n) = 2^n - 1$ é a fórmula fechada para a recorrência:

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

Demonstração

- **Caso base:** Para $n = 0$, $2^0 - 1 = 0 = T(0)$.
- **Hipótese:** Assuma que $T(k) = 2^k - 1$ (para $n = k$). Queremos mostrar que quando $n = k + 1$, $T(k+1) = 2^{k+1} - 1$.
- **Passo:** Usando a hipótese para $T(k)$ na recorrência, temos:

$$\begin{aligned} T(k+1) &= 2T((k+1)-1) + 1 \\ &= 2.T(k) + 1 \\ &= 2.(2^k - 1) + 1 \\ &= 2^{k+1} - 1. \end{aligned}$$

Recorrências e análise assintótica

Analisamos a complexidade do Merge-sort negligenciando alguns detalhes técnicos. Uma recorrência mais precisa seria:

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{se } n > 1 \end{cases}$$

Simplificações (a serem usadas com cuidado)

Por conveniência:

- Ignoramos o fato de que n deveria ser inteiro (omissão de pisos e tetos).
- Mudanças no valor de $T(1)$ alteram o valor exato da recorrência (tipicamente não mais que um fator constante). Omissão da condição limite.
- Normalmente a recursão acima é apresentada como:

$$T(n) = 2T(n/2) + \Theta(n).$$

Método da substituição

Consiste em duas etapas:

1. Adivinhe a forma da solução.
2. Use indução matemática para encontrar as constantes e mostrar que a solução funciona.

Método da substituição

Consiste em duas etapas:

1. Adivinhe a forma da solução.
2. Use indução matemática para encontrar as constantes e mostrar que a solução funciona.

Um exemplo

Podemos usar o método da substituição para provar limites superiores e inferiores de uma recorrência.
Determine um limite superior para $T(n) = 2T(n/2) + n$.

Solução para o exemplo: $T(n) = 2T(n/2) + n$

1. Suponha que $T(n) = O(n \lg n)$ (**chute**).
2. Use indução matemática.

Precisamos mostrar que $T(n) \leq cn \lg n$, com $c > 0$

- **Hipótese:** Assumimos que $T(k) \leq ck \lg k$ para todo $k < n$, em particular para $k = n/2$ (isto é, $T(n/2) \leq c(n/2)\lg(n/2)$).
- **Passo:**

$$
\begin{aligned}
T(n) &= 2T(n/2) + n \\
&\leq 2(c(n/2)\lg(n/2)) + n \\
&= cn\lg(n/2) + n \\
&= cn(\lg n - \lg 2) + n \\
&= cn\lg n - cn + n \\
&\leq cn \lg n && \text{(para } c \geq 1\text{).}
\end{aligned}
$$

Solução para o exemplo: $T(n) = 2T(n/2) + n$

Considerações sobre o caso base

Considerando $T(n) \leq cn \lg n$:

- Devemos escolher um c que atenda o caso base também.
- Problema: suponha que $T(1) = 1$ para a recorrência, então $T(1) \leq cn \lg n = c1 \lg 1 = 0 \neq T(1)$ (consequentemente nossa prova por indução falhou).
- Para contornar o problema, definiremos outro caso base (que não dependa de 1), pois a notação assintótica requer que isto seja válido para um $n \geq n_0$.
- Considere $n_0 = 2$. Pela recorrência, temos que $T(2) = 4$ e $T(3) = 5$. Podemos então definir um valor para c tal que $T(2) \leq c2 \lg 2$ e $T(3) \leq c3 \lg 3$. Assumindo $c \geq 2$ concluimos a demonstração.

Sua vez

1. Mostre que $T(n) = T(n-1) + n$ é $O(n^2)$.
2. Mostre que $T(n) = 4T(n/2) + n^2$ é $O(n^2 \lg n)$.

Método da substituição

Hipótese mais forte

- Considere a recorrência: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$.
- Vamos supor que $T(n) = O(n)$.
- Aplicando substituição não conseguimos chegar na forma exata da indução (isto é, não chegamos em $T(n) \leq cn$).
- Neste caso, devemos usar uma hipótese indutiva mais forte – $T(n) \leq cn - b$, onde $b \geq 0$ é uma constante.

Exercício

Considere a recorrência: $T(n) = 8T(n/2) + \Theta(n^2)$. Mostre que $T(n) = O(n^3)$.