CI165 — Recorrências 2 - Adivinhando Soluções de recorrências

André Vignatti

31 de julho de 2014

Método da iteração

- Serve para **adivinhar** a resposta!
- Precisa fazer muitas contas!
- Precisa conhecer limitantes para várias somatórias.

Ideia

"Abrir" (iterar) a recorrência e escrevê-la como uma somatória de termos que dependem apenas de n e das condições iniciais.

Diferença de CI237: agora, em CI165, as recorrências tem soluções **assintóticas**.

- Ou seja, não precisamos da solução **exata** (Eeeeebbaaa!!)

Método da iteração

Considere a recorrência

$$
\begin{aligned}
T(n) &= b && \text{para } n \leq 3, \\
T(n) &= 3T(\lfloor n/4 \rfloor) + n && \text{para } n \geq 4.
\end{aligned}
$$

Iterando a recorrência obtemos

$$
\begin{aligned}
T(n) &= n + 3T(\lfloor n/4 \rfloor) \\
&= n + 3(\lfloor n/4 \rfloor + 3T(\lfloor n/16 \rfloor)) \\
&= n + 3(\lfloor n/4 \rfloor + 3(\lfloor n/16 \rfloor + 3T(\lfloor n/64 \rfloor))) \\
&= n + 3\lfloor n/4 \rfloor + 9\lfloor n/16 \rfloor + 27T(\lfloor n/64 \rfloor).
\end{aligned}
$$

Certo, mas quando devo parar?
O i-ésimo termo da série é $3^i \lfloor n/4^i \rfloor$. Ela termina quando $\lfloor n/4^i \rfloor \leq 3$, ou seja, $i \geq \log_4 n$.

Método da iteração

Como $\lfloor n/4^i \rfloor \leq n/4^i$ temos que

$$\begin{aligned}
T(n) &\leq n + 3n/4 + 9n/16 + 27n/64 + \cdots + 3^j b \\
T(n) &\leq n + 3n/4 + 9n/16 + 27n/64 + \cdots + d.3^{\log_4 n} \\
&= n.(1 + 3/4 + 9/16 + 27/64 + \cdots) + dn^{\log_4 3} \\
&\leq n \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i + dn^{\log_4 3} \\
&= 4n + dn^{\log_4 3}
\end{aligned}$$

pois $3^{\log_4 n} = n^{\log_4 3}$ e $\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}$ para $0 < q < 1$.

Como $\log_4 3 < 1$ segue que $n^{\log_4 3} \in O(n)$ e logo, $T(n) \in O(n)$.

Método de iteração

Dica: removendo pisos e tetos

- Lidar com pisos e tetos é chato!
- Pode-se remover pisos e tetos se supor que a recorrência é definida apenas para potências de um número, por exemplo, $n = 4^i$.
  - Mas a recorrência deve ser provada para todo $\mathbb{N}$.
- Então,
  (1) **adivinha** um chute usando potências,
  (2) mas **prova-se** o chute para todo $\mathbb{N}$ usando indução.

Método de iteração

Vamos provar usando substituição (indução)

$$\begin{aligned} T(n) &= b && \text{para } n \leq 3, \\ T(n) &= 3T(\lfloor n/4 \rfloor) + n && \text{para } n \geq 4. \end{aligned}$$

Chuto que $T(n) \leq cn$.

$$\begin{aligned} T(n) &= 3T(\lfloor n/4 \rfloor) + n \\ &\leq 3c\lfloor n/4 \rfloor + n \\ &\leq 3c(n/4) + n \\ &\leq cn \end{aligned}$$

onde a última desigualdade vale se $c \geq 4$.

Árvore de recorrência

- Permite visualizar melhor o que acontece quando a recorrência é iterada.
- É mais fácil organizar as contas.
- Útil para recorrências de algoritmos de divisão-e-conquista.

Árvore de recorrência

Considere a recorrência

$$\begin{aligned} T(n) &= \Theta(1) && \text{para } n = 1, 2, 3, \\ T(n) &= 3T(\lfloor n/4 \rfloor) + cn^2 && \text{para } n \geq 4, \end{aligned}$$

onde $c > 0$ é uma constante.

Costuma-se (CLRS) usar a notação $T(n) = \Theta(1)$ para indicar que $T(n)$ é uma constante.

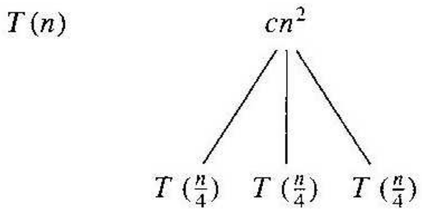
Simplificação

Vamos supor que a recorrência está definida apenas para potências de 4

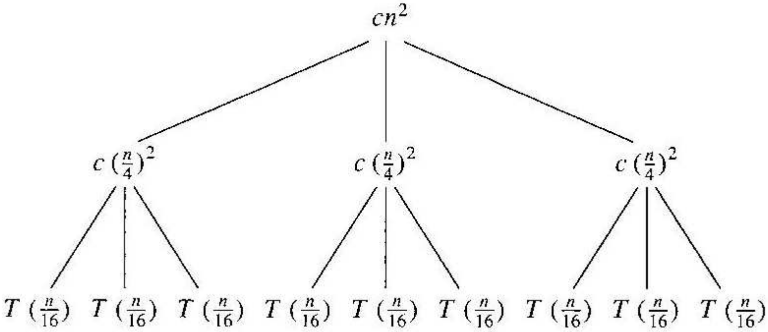
$$
\begin{aligned}
T(n) &= \Theta(1) && \text{para } n = 1, \\
T(n) &= 3T(n/4) + cn^2 && \text{para } n = 4, 16, \ldots, 4^i, \ldots.
\end{aligned}
$$

Isto permite descobrir mais facilmente a solução. Depois usamos o método da substituição para formalizar.

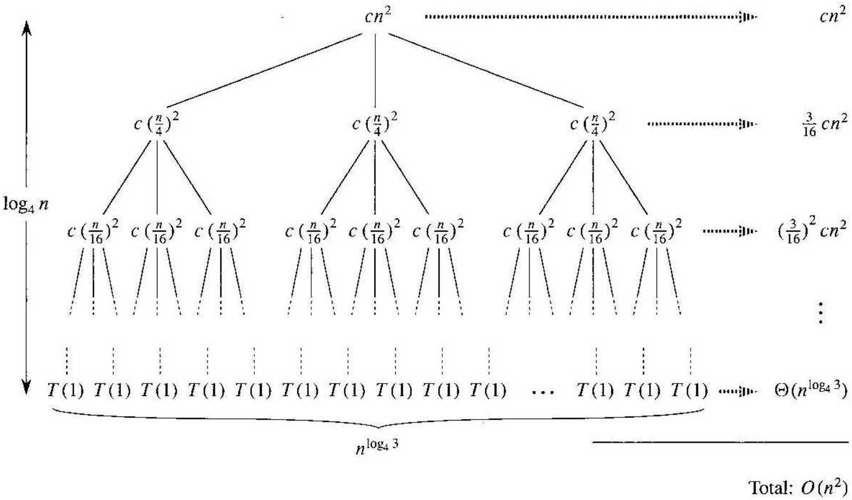
Árvore de recorrência



Árvore de recorrência



Árvore de recorrência



Árvore de recorrência

- O número de níveis é $\log_4 n + 1$.
- No nível i o tempo gasto (sem contar as chamadas recursivas) é $(3/16)^i cn^2$.
- No último nível há $3^{\log_4 n} = n^{\log_4 3}$ folhas. Como $T(1) = \Theta(1)$ o tempo gasto é $\Theta(n^{\log_4 3})$.

Logo,

$$\begin{aligned}
T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \left(\frac{3}{16}\right)^3 cn^2 + \cdots + \\
&\quad + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\
&= cn^2 \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3}) \\
&\leq cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3}) = \frac{16}{13}cn^2 + \Theta(n^{\log_4 3}),
\end{aligned}$$

e $T(n) \in O(n^2)$.

Árvore de recorrência

Mas $T(n) \in O(n^2)$ é realmente a solução da recorrência original?

Com base na árvore de recorrência, chutamos que $T(n) \leq dn^2$ para alguma constante $d > 0$.

$$\begin{aligned} T(n) &= 3T(\lfloor n/4 \rfloor) + cn^2 \\ &\leq 3d\lfloor n/4 \rfloor^2 + cn^2 \\ &\leq 3d(n/4)^2 + cn^2 \\ &= \frac{3}{16}dn^2 + cn^2 \\ &\leq dn^2 \end{aligned}$$

onde a última desigualdade vale se $d \geq (16/13)c$.
(Yeeesssss!)

Resumo

- O número de nós em cada nível da árvore é o número de chamadas recursivas.
- Em cada nó indicamos o “tempo” ou “trabalho” gasto naquele nó que **não** corresponde a chamadas recursivas.
- Na coluna mais à direita indicamos o tempo total naquele nível que **não** corresponde a chamadas recursivas.
- Somando ao longo da coluna determina-se a solução da recorrência.

Vamos tentar juntos?

Eis um exemplo.

Vamos resolver a recorrência

$$\begin{array}{rcll} T(n) & = & 1 & \text{para } n = 1, 2, \\ T(n) & = & T(n/3) + T(2n/3) + n & \text{para } n \geq 3. \end{array}$$

Qual é a solução da recorrência?

Resposta: $T(n) \in O(n \lg n)$. (Resolvido em aula)

Recorrências com O à direita (CLRS)

Uma “recorrência”

$$T(n) = \Theta(1) \qquad \text{para } n = 1, 2,$$
$$T(n) = 3T(\lfloor n/4 \rfloor) + \Theta(n^2) \qquad \text{para } n \geq 3$$

representa todas as recorrências da forma

$$T(n) = a \qquad \text{para } n = 1, 2,$$
$$T(n) = 3T(\lfloor n/4 \rfloor) + bn^2 \qquad \text{para } n \geq 3$$

onde a e $b > 0$ são constantes.

As soluções exatas dependem dos valores de a e b, mas estão todas na mesma classe Θ.

A “solução” é $T(n) = \Theta(n^2)$, ou seja, $T(n) \in \Theta(n^2)$.

As mesmas observações valem para as classes O, Ω, o, ω.

Recorrência do Mergesort

Podemos escrever a recorrência de tempo do Mergesort da seguinte forma

$$\begin{aligned} T(1) &= \Theta(1) \\ T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) \qquad \text{para } n \geq 2. \end{aligned}$$

A solução da recorrência é $T(n) = \Theta(n \lg n)$.

A prova é essencialmente a mesma do primeiro exemplo. (Exercício!)