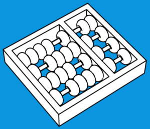
Análise de algoritmos recursivos

MO417 - Complexidade de Algoritmos I

6

03/24



Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

UNICAMP

Recursão, aqui vamos de novo!

Recursão, aqui vamos de novo!

Recursão, aqui vamos de novo!

Recursão, aqui vamos de novo!

Recursão, aqui vamos de novo!

Recursão, aqui vamos de novo!

Última aula

O que foi visto?

- Método de substituição.
- Forma de encontrar constantes c e n_0 para a prova por indução.

O que falta?

- Forma de adivinhar uma solução da recorrência:
  - Iteração.
  - Árvore de recorrência.
- Uma forma geral para várias recorrências.
- Prova de correção de algoritmos recursivos.

Adivinhando uma solução

Não há receita genérica para adivinhar soluções.

- A experiência é um fator importante.
- Normalmente, recorrências similares têm a mesma ordem de crescimento.

Exemplo

Resolva a recorrência

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

- Ela muito parecida com a anterior.
- Então chutamos $T(n) \in \Theta(n \log_2 n)$.
- Mostre isso como exercício.

Adivinhando uma solução (cont)

Exemplo

Resolva a recorrência

$$T(n) = \begin{cases} 1 & se\ n = 1 \\ 2T(\lfloor n/2 \rfloor + 17) + n & se\ n \geq 2 \end{cases}$$

- Ela parece mais difícil por causa do termo 17.
- Intuitivamente, parece que $T(\lfloor n/2 \rfloor) \approx T(\lfloor n/2 \rfloor + 17)$
- Chutamos que $T(n) \in \Theta(n \log_2 n)$.
- Mostre isso como exercício.

Sutilezas

Algumas vezes é necessário fortalecer a hipótese de indução.

Exemplo

Resolva a recorrência

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

Chutamos que $T(n) \leq cn$ para alguma constante c.

$$\begin{aligned} T(n) &= \mathbf{T(\lceil n/2 \rceil)} + \mathbf{T(\lfloor n/2 \rfloor)} + 1 \\ &\leq \mathbf{c\lceil n/2 \rceil} + \mathbf{c\lfloor n/2 \rfloor} + 1 \\ &= cn + 1. \end{aligned}$$

- Não deu certo.
- Mas de fato $T(n) \in \Theta(n)$!

Fortalecendo a hipótese

- Vamos fazer uma hipótese de indução um pouco diferente.
- Vamos mostrar que $T(n) \le cn - b$ para $c > 0$ e $b > 0$:

$$\begin{aligned}
T(n) &= \mathbf{T}(\lceil \mathbf{n}/\mathbf{2} \rceil) + \mathbf{T}(\lfloor \mathbf{n}/\mathbf{2} \rfloor) + 1 \\
&\le \mathbf{c}\lceil \mathbf{n}/\mathbf{2} \rceil - \mathbf{b} + \mathbf{c}\lfloor \mathbf{n}/\mathbf{2} \rfloor - \mathbf{b} + 1 \\
&= cn - 2b + 1 \\
&\le cn - b.
\end{aligned}$$

- Para a última de desigualdade, basta escolher $b \ge 1$.
- Isso mostra o passo indutivo.

Método da iteração

Ideia:

1. Expandir a recorrência iterativamente.
2. Reescrevê-la como uma somatória em termos de n.

Não é necessário adivinhar a resposta!

- Mais é um pouco mais trabalhoso.
- É útil conhecer limitantes de somatórias.
- Nem sempre é trivial encontrar um termo geral da soma.

Exemplo

Exemplo

Encontre uma fórmula fechada para

$$T(n) = \begin{cases} b & se\ n \leq 3 \\ 3T(\lfloor n/4 \rfloor) + n & se\ n \geq 4 \end{cases}$$

Iterando a recorrência:

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

Identificamos:

- O termo geral da soma: $3^i n/4^i$.
- O argumento da função: $\lfloor n/4^k \rfloor$.

Encontrando a função

Quantas vezes iteramos?

- Suponha que iteramos k vezes.
- Paramos quando $\lfloor n/4^k \rfloor \leq 3$, i.e., quando $k+1 > \log_4 n \geq k$

A soma completa é:

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

Como $\log_4 3 < 1$, concluímos que $T(n) \in O(n)$.

Simplificando

Podemos supor que $n = 4^k$:

- As contas ficam mais simples.
- Não prova o caso geral.
- Mas obtemos um bom **CHUTE**.
- E verificamos com o método da substituição.

Exercício: verifique que $T(n) = \Theta(n)$.

Árvore de recorrência

Ideia:

1. Crie uma árvore de recorrência:
   - Os nós representam os termos independentes.
   - Os filhos representam as subfunções recorrentes.
2. Somamos os termos de cada nível da árvore.
3. Depois somamos todos os níveis.

Vantagens:

- Útil quando há vários termos recorrentes.
- É mais fácil organizar as contas.

Exemplo

Exemplo

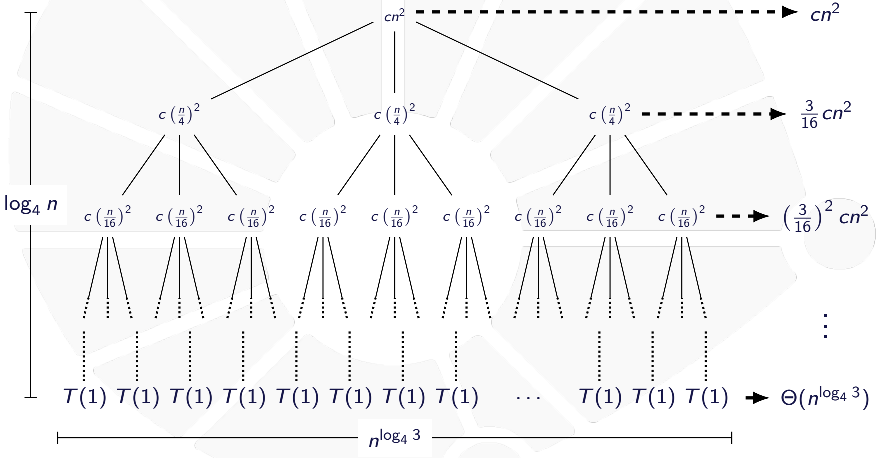
Encontre uma fórmula fechada para

$$T(n) = \begin{cases} 1 & se\ n \leq 3 \\ 3T(\lfloor n/4 \rfloor) + cn^2 & se\ n \geq 4 \end{cases}$$

Simplificando:

- Queremos obter uma fórmula fechada apenas.
- De novo, supomos que $n = 4^k$.
- Depois, verificamos com o método da substituição.

Árvore de recorrência



Total: $\Theta(n^2)$

Somando os termos

- A árvore tem altura $\log_4 n$.
- A soma de cada nível interno é $\left(\frac{3}{16}\right)^i cn^2$.
- O número de **folhas** é $3^{\log_4 n} = n^{\log_4 3}$.

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

Concluímos que $T(n) \in O(n^2)$.

Outro exemplo

Exemplo

Encontre uma fórmula fechada para

$$T(n) = \begin{cases} 1 & \text{se } n \leq 2 \\ T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n & \text{se } n \geq 3 \end{cases}$$

Observe que:

- Os termos recorrentes são diferentes.
- A árvore não será completa.

Exercício: Use a árvore para chutar a fórmula da recorrência e substituição para provar que é correta.

Recorrências com notação assintótica

- Escrevemos:

$$T(n) = \begin{cases} \Theta(1) & \text{se } n \leq 2 \\ 3T(\lfloor n/4 \rfloor) + \Theta(n^2) & \text{se } n \geq 3 \end{cases}$$

- Para representar a **família de recorrências**:

$$T(n) = \begin{cases} a & \text{se } n \leq 2 \\ 3T(\lfloor n/4 \rfloor) + f(n) & \text{se } n \geq 3 \end{cases}$$

onde a é uma constante e $f(n)$ está em $\Theta(n^2)$.

- Todas elas têm a mesma solução assintótica.

- A mesma coisa vale as outras classes de função.

Cuidados com a notação assintótica

Exemplo

$$T(n) = \begin{cases} 1 & se\ n = 1 \\ 2T(\lfloor n/2 \rfloor) + n & se\ n \geq 2, \end{cases}$$

- O uso descuidado da notação assintótica leva a **ERROS**.
- Já sabemos que $T(n) = \Theta(n \log n)$.
- Mas vamos "provar" que $T(n) = O(n)$!

Cuidados com a notação assintótica

Vamos mostrar que $T(n) \leq cn$ para alguma constante $c > 0$:

$$\begin{aligned} T(n) &= 2T(\lfloor n/2 \rfloor) + n \\ &\leq 2c\lfloor n/2 \rfloor + n \\ &\leq cn + n \\ &= O(n) \qquad \longleftarrow \textbf{ERRADO!} \end{aligned}$$

Qual é o erro?

- A nossa afirmação é que $T(n) \leq cn$.
- Mas mostramos que $T(n) \leq (c+1)n$.
- Não concluímos o passo indutivo.

Teorema Master

Veremos um teorema para resolver recorrências da forma:

$$T(n) = aT(n/b) + f(n).$$

- Os valores a e b são constantes.
- Também vale se substituirmos n/b por $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$.
- O teorema **NÃO** se aplica a todas recorrências dessa forma.

Teorema Master

Teorema (Teorema Master)

Sejam $a \geq 1$ e $b > 1$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida para os inteiros não negativos pela relação de recorrência:

$$T(n) = aT(n/b) + f(n).$$

Então, $T(n)$ tem o seguinte limitante assintótico:

1. *Se $f(n) \in O(n^{\log_b a-\epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) \in \Theta(n^{\log_b a})$.*
2. *Se $f(n) \in \Theta(n^{\log_b a})$, então $T(n) \in \Theta(n^{\log_b a} \log n)$.*
3. *Se $f(n) \in \Omega(n^{\log_b a+\epsilon})$ para alguma constante $\epsilon > 0$ e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e todo n suficientemente grande, então $T(n) \in \Theta(f(n))$.*

Exemplos de Recorrências

Exemplo (Aplicação do Teorema Master)

- *Caso 1:*
  $T(n) = 9T(n/3) + n.$
  $T(n) = 4T(n/2) + n \log n.$

- *Caso 2:*
  $T(n) = T(2n/3) + 1.$
  $T(n) = 2T(n/2) + (n + \log n).$

- *Caso 3:*
  $T(n) = T(3n/4) + n \log n.$

Exemplos de Recorrências

Exemplo (NÃO aplicação do Teorema Master)

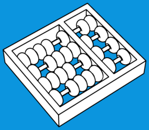
- $T(n) = T(n-1) + n$.
- $T(n) = T(n-a) + T(a) + n$, $(a \geq 1$ *inteiro*$)$.
- $T(n) = T(\alpha n) + T((1-\alpha)n) + n$, $(0 < \alpha < 1)$.
- $T(n) = T(n-1) + \log n$.
- $T(n) = 2T(\frac{n}{2}) + n \log n$.

Análise de algoritmos recursivos

MO417 - Complexidade de Algoritmos I

6

03/24



UNICAMP

Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br