

# Relações de Recorrência: Método da Substituição

Análise de Algoritmos – Ciência da Computação



Prof. Daniel Saad Nogueira  
Nunes

IFB – Instituto Federal de Brasília,  
Campus Taguatinga



# Sumário

---

- 1 Método da Substituição
- 2 Sutilezas
- 3 Mudança de Variáveis
- 4 Árvores de recursão



# Sumário

---

## 1 Método da Substituição



# Método da Substituição

---

## Método da Substituição

O método da substituição se baseia em:

- 1 “Chutar” a forma da solução;
- 2 Verificar que o chute está correto através da indução matemática e escolhas de constantes apropriada;
- 3 Ajustar as cotas superiores e inferiores de modo a conseguir uma cota justa;



# Método da Substituição

---

## Exemplo

Determine uma cota superior para:

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

❶ Chute:  $T(n) \in O(n \lg n)$ . Isto é:  $T(n) \leq cn \lg n$ ;

❷ Indução:

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \\ &\leq cn \lg(n/2) + n \\ &= cn(\lg(n) - 1) + n \\ &= cn \lg(n) - cn + n \\ &\leq cn \lg n \quad \diamond c \geq 1 \end{aligned}$$



# Método da Substituição

---

## Exemplo

- Verificação do caso base.

$$1 \not\leq c(1 \lg 1)$$

- A verificação do caso base falhou.
- Como estamos realizando uma análise assintótica, não precisamos provar para  $n_0 = 1$ , só precisamos mostrar que a indução vale para todo  $n \geq n_0$  suficientemente grande. Vamos mostrar que a propriedade vale para todo  $n \geq n_0 > 1$ .



# Método da Substituição

---

## Exemplo

- Verificação do caso base quando  $n_0 > 1$ .
- Pela natureza da recorrência, qualquer valor de  $n > 1$  passará por  $T(2)$  ou  $T(3)$  nas chamadas recursivas. Vamos verificar que a propriedade vale para  $T(2)$  e  $T(3)$

$$T(2) = 2T(1) + 2 \leq c2 \lg 2, \quad \diamond c \geq 2$$

$$T(3) = 2T(1) + 3 \leq c3 \lg 3, \quad \diamond c \geq 2$$

- Mostramos que a indução funciona para todo  $n > 1$  com a constante  $c \geq 2$  escolhida.



# Método da Substituição

---

## Exercício

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

- Será que  $\Theta(n \lg n)$  é uma cota justa? Ajuste!





# Método da Substituição

---

## Método da Substituição

- Como dar um bom chute? Experiência. Muitas relações de recorrência são similares:

$$T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$$

- Pisos e tetos não são importantes na maioria das vezes e não afetam o comportamento assintótico da recorrência. Por isso, muitas das vezes, as relações de recorrência, em uma primeira análise, são expressas sem pisos e tetos. Caso valha a pena, fazemos uma análise mais minuciosa posteriormente.
- O mesmo pode ser dito do caso base.



# Método da Substituição

---

## Método da Substituição

- Ajuste das cotas: Podemos superestimar cotas superiores e subestimar cotas inferiores e ir ajustando progressivamente para deixá-las justas. Exemplo:  $O(n^2)$  e  $\Omega(n)$  para a recorrência:

$$T(n) = 2T(n/2 + 17) + n$$



# Sumário

---

## 2 Sutilezas



# Método da Substituição

---

## Reforçando a Hipótese de Indução

- Muitas das vezes, nosso chute pode estar correto, mas mesmo assim, não obtemos a resposta na indução matemática.
- Neste casos, podemos reforçar nossa hipótese de indução em uma mais difícil, mas que acaba facilitando a demonstração.



# Método da Substituição

---

## Exemplo

Ache uma cote superior para:

$$T(n) = 2T(n/2) + 1$$

Chute:  $T(n) \leq cn$ :

$$\begin{aligned} T(n) &\leq 2(cn/2) + 1 \\ &= cn + 1 \\ &\not\leq cn \end{aligned}$$



# Método da Substituição

---

## Exemplo

Ache uma cote superior para:

$$T(n) = 2T(n/2) + 1$$

A conta não fecha, apesar do chute inicial estar correto! Vamos reforçar a hipótese de indução:  $T(n) \leq cn - b$ .



# Método da Substituição

---

## Exemplo

Ache uma cote superior para:

$$T(n) = 2T(n/2) + 1$$

Chute:  $T(n) \leq cn - b$ :

$$\begin{aligned} T(n) &\leq 2(cn/2 - b) + 1 \\ &= cn - 2b + 1 \\ &\leq cn - b \quad \diamond \quad \forall n > n_0 \text{ suficientemente grande} \wedge b \geq 1 \end{aligned}$$



# Sumário

---

## 3 Mudança de Variáveis





# Método da Substituição

---

## Mudança de Variáveis

- Para simplificar a resolução de recorrências, podemos fazer mudanças de variáveis.



# Método da Substituição

---

## Exemplo

Ache uma cota superior para:

$$T(n) = 2T(\sqrt{n}) + \lg n$$

Defina  $m := \lg n$ .

$$T(2^m) = 2T(2^{m/2}) + m$$

Defina  $S(m) = T(2^m)$ .

$$S(m) = 2S(m/2) + m$$

Sabemos que  $S(m) \in O(m \lg m)$ . Trocando de volta as variáveis, temos:  $S(m) \in O(m \lg m) = O(\lg n \lg \lg n)$ .



# Sumário

---

## 4 Árvores de recursão



# Árvores de recursão

---

- O método da substituição fornece uma forma simples de demonstrar cotas superiores e inferiores para relações de recorrência.
- Contudo, para podermos utilizar este método, precisamos de uma estimativa, um chute de uma cota justa.
- Pode ser difícil dar uma boa estimativa.
- Para obtê-la, podemos utilizar um mecanismo conhecido como **árvore de recursão**.



# Árvore de recursão

---

- Árvores de recursão são utilizadas para obter uma estimativa para a cota superior ou inferior para então aplicarmos o método da substituição.
- Como o interesse é gerar uma estimativa, podemos ser um pouco mais grosseiros com relação a certos cálculos, já que a estimativa será verificada formalmente em um momento posterior.



# Árvore de recursão

---

## Exemplo

Determine uma boa estimativa de cota superior para a relação de recorrência:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3T(n/4) + \Theta(n^2), & n > 1 \end{cases}$$



# Árvore de recursão

---

- A ideia aqui é gerar uma árvore em que cada nó interno represente o custo de cada subproblema.
- O custo por nível da árvore corresponde à soma dos custos dos subproblemas daquele nível.
- O custo total por sua vez corresponde à soma dos custos de todos os níveis da árvore.



## Árvore de recursão

---

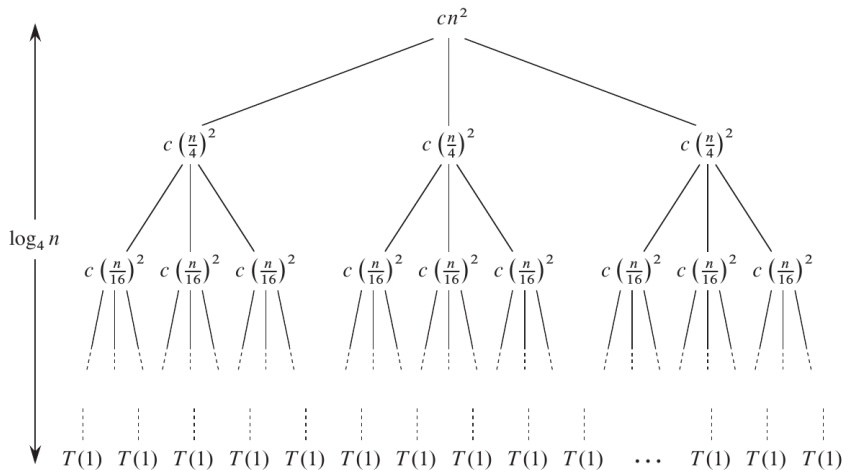
$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3T(n/4) + \Theta(n^2), & n > 1 \end{cases}$$

- Como a ideia é obter uma estimativa apenas, vamos assumir que  $n$  é potência de 4.
- Vamos tentar obter primeiro uma estimativa para cota superior.
- Cada problema (nó) terá custo de uma função em  $\Theta(n^2) \leq cn^2$ .
- Além disso, cada problema gerará 3 subproblemas de tamanho  $\frac{1}{4}$  do original.
- Como  $n$  é dividido por 4 a cada nível, a árvore terá altura  $\log_4 n$ .





# Árvore de recursão





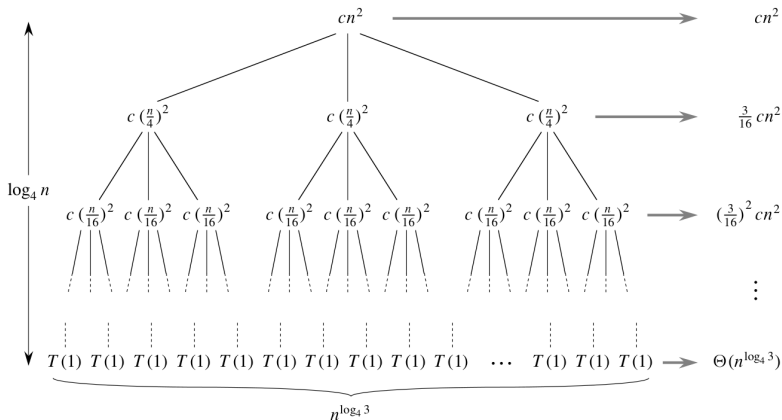
# Árvore de recursão

---

- O número de folhas de uma árvore de grau 3 perfeita é  $3^h$ , em que  $h$  é sua altura. Como  $h = \log_4 n$ , temos que o número de folhas é  $3^{\log_4 n} = n^{\log_4 3}$ . Como cada folha tem custo  $\Theta(1)$ , o custo total deste nível é  $\Theta(n^{\log_4 3})$ .
- O custo de cada um dos demais níveis por sua vez é:  $(\frac{3cn}{16})^2$ .



# Árvore de recursão





## Árvore de recursão

---

$$\begin{aligned}T(n) &= \sum_{i=0}^{\log_4 n} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \quad \diamond (3^{\log_4 n}) \text{ folhas} \\&< \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\&= \frac{1}{1 - \left(\frac{3}{16}\right)} cn^2 + \Theta(n^{\log_4 3}) \quad \diamond \text{ Soma da PG} \\&= \frac{16}{13} cn^2 + \Theta(n^{\log_4 3}) \in O(n^2)\end{aligned}$$



# Árvore de recursão

---

- Determinamos que uma boa estimativa para cota superior é  $O(n^2)$ .
- Só precisamos agora usar essa estimativa para provar o que queremos via método da substituição.



# Árvore de recursão

---

① Chute:  $T(n) \in O(n^2)$ . Isto é:  $T(n) \leq cn^2$ ;

② Indução:

$$\begin{aligned} T(n) &\leq 3c(n/4)^2 + dn^2 \\ &\leq \frac{3cn^2}{16} + dn^2 \\ &\leq cn^2 \quad \diamond c \geq \frac{16d}{13} \end{aligned}$$



# Árvore de recursão

---

- Como a relação de recorrência já possui um termo  $\Theta(n^2)$  na sua forma geral, não precisamos provar a cota inferior, visto que o custo já tem que ser no mínimo  $\Omega(n^2)$ .
- Provamos a cota justa de  $\Theta(n^2)$ .