① 4ª Aula

Notações Chão e Teto

O chão (ou piso) de um número não negativo x é o único número natural i tal que $i \leq x < i+1$. O chão de x é denotado por $\lfloor x \rfloor$.

O teto de x é o único número j natural tal que $j-1 < x \leq j$. O teto de x é denotado por $\lceil x \rceil$.

②

Divisão e Conquista:

Dada uma instância do problema, dividi-lo em instâncias menores, resolvê-los recursivamente e, a partir de suas soluções, obter a solução da instância original.

Merge-Sort:

Divide-se a sequência em duas partes, ordena-se cada uma delas recursivamente e intercala-se as duas sequências ordenadas.

③

```
Merge-Sort (A, p, r)                    Custo
    if p<r then                         Θ(1)
        q = ⌊(p+r)/2⌋                   Θ(1)
        Merge-Sort (A, p, q)            T(⌈n/2⌉)
        Merge-Sort (A, q+1, r)          T(⌊n/2⌋)
        Merge (A, p, q, r)              Θ(n)
```

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

Resolver recorrências é importante para obter fórmulas fechadas para a complexidade de algoritmos recursivos.

④ Métodos para resolver recorrências

→ Substituição: Arriscamos um palpite para um limite e então usamos indução matemática para provar que nosso palpite estava correto.

→ Árvore de recursão: Convertemos a recorrência em uma árvore e usamos técnicas para limitar somatórios com intuito de resolver a recorrência.

→ Método Mestre: Para recorrências da forma $T(n) = a\,T(n/b) + f(n)$, com $a \geq 1$ e $b > 1$.

⑤

Lembre-se que:

A meta não é determinar as soluções exatas de uma recorrência, mas encontrar uma solução $g(n)$ tal que $T(n) = \Theta(g(n))$.

Metodo de substituição

Vamos resolver a recorrência que expressa o tempo de execução do algoritmo Merge-Sort

Primeiro, simplificar a recorrência:

⑥

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

Polpite: $T(n) = O(n \lg n)$. Ou seja, $T(n) \leq c\, n \lg n$, para alguma constante c quando $n \geq n_0$.

Base: $T(2) = T(1) + T(1) + 2 = 4 \leq 3 \cdot 2 \cdot \lg 2 = 6$

Hipótese: $T(m) \leq c m \lg m$, para todo $m < n$.

Passo indutivo:

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

⑦

$$\leq c\lceil n/2\rceil \lg \lceil n/2\rceil + c\lfloor n/2\rfloor \lg \lfloor n/2\rfloor + n$$
$$\leq c\lceil n/2\rceil \lg n + c\lfloor n/2\rfloor (\lg n - 1) + n$$
$$= c(\lceil n/2\rceil + \lfloor n/2\rfloor)\lg n - c\lfloor n/2\rfloor + n$$
$$= cn\lg n - c\lfloor n/2\rfloor + n$$
$$\leq cn\lg n$$

A inequação $n - c\lfloor n/2\rfloor \leq cn\lg n$ é válido para $c \geq 3$, $n \geq 2$.

———— // ————

⑧

Resolver a recorrência:

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

Palpite: $T(n) = O(n)$

Base okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Hipotese: $T(m) \leq cm$, para todo $m < n$.

Passo indutivo:

$$T(n) = T(\lfloor n/2\rfloor) + T(\lceil n/2\rceil) + 1$$
$$\leq c\lceil n/2\rceil + c\lfloor n/2\rfloor + 1$$
$$= cn + 1 \ (?)$$

9

Solução para (?).

Substituir a hipótese por $T(n) \leq cn - b$, com $b \geq$

$T(n) \leq cn - b$ permite as escolhas $c = 2$, $b = 1$ e $n_0 = 1$.

Método da árvore de recursão

Esquema geral:

→ Simplifique a fórmula de recorrência

→ Itere a recorrência visualmente como uma árvore.

10

→ Indique em cada nó o custo correspondente às chamadas recursivas

→ Calcular para cada nível da árvore o custo que não corresponde à chamadas recursivas.

→ A solução da recorrência é a soma de todos os custos em cada nível.

Importante: Depois de encontrar a solução deve-se aplicar o método de substituição!

(11) Resolver a recorrência a seguir para uma constante $c > 0$:

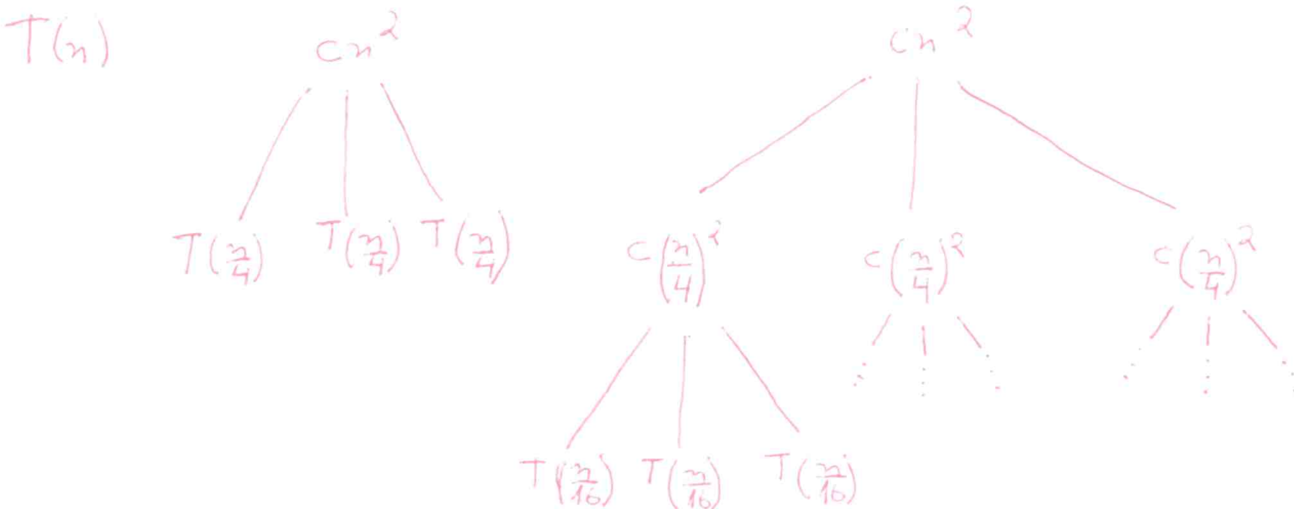
$$T(n) = \begin{cases} \Theta(1) & , \text{se } n \leq 3 \\ 3T(\lfloor n/4 \rfloor) + \Theta(n^2) & , \text{se } n \geq 4 \end{cases}$$

Simplifiquemos a recorrência:

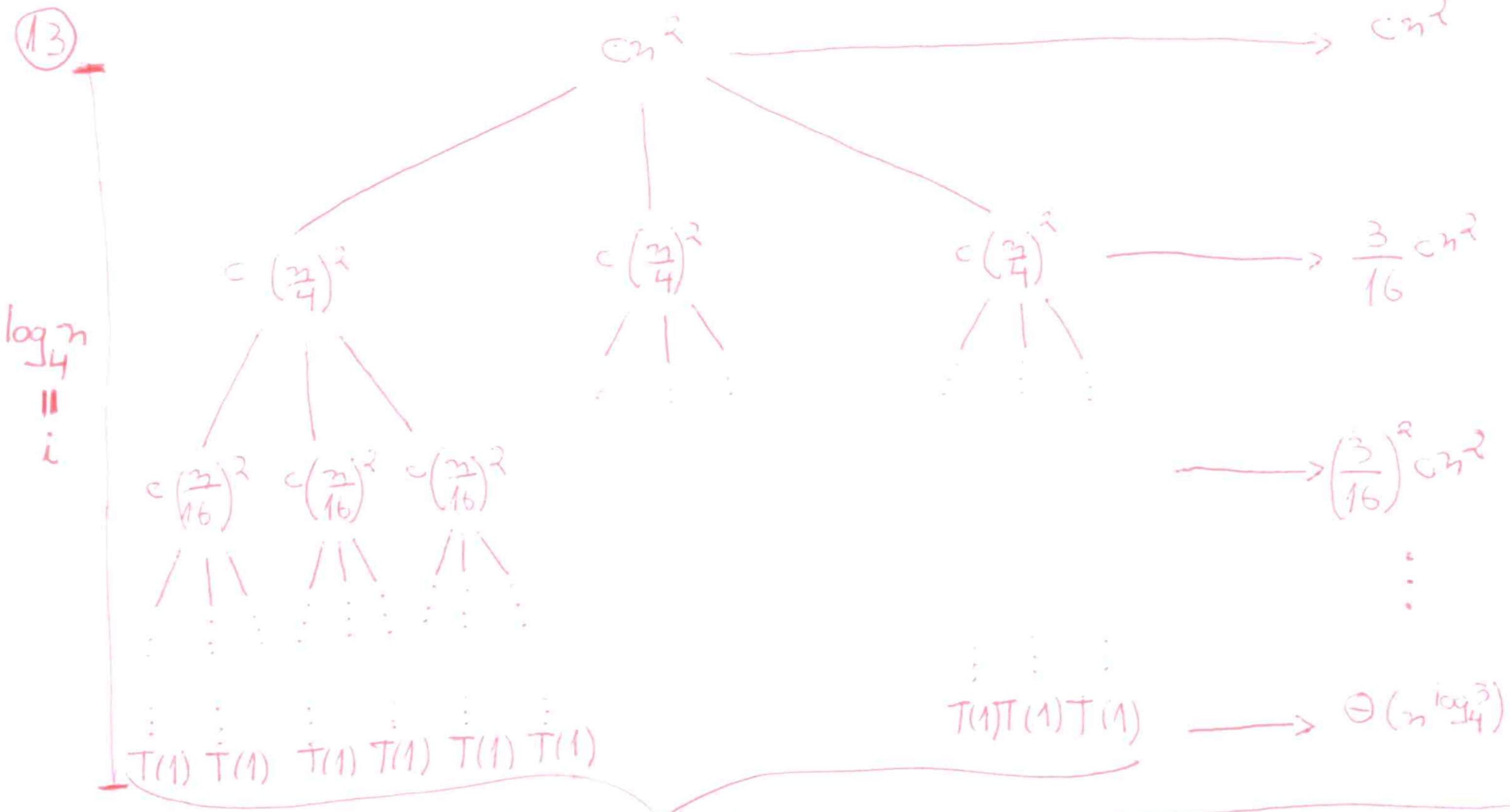
$$T(n) = \begin{cases} \Theta(1) & , \text{se } n = 1 \\ 3T(n/4) + cn^2 & , \text{se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Visualizemos a árvore:

(12)



(13)



$3^{\log_4 n} = n^{\log_4 3}$

Total: $O(n^2)$

3^i = número de nós na profundidade $i = 0, 1, \ldots, \log_4 n - 1$

(14)

$$T(n) = cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3})$$

$$= cn^2 \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3})$$

$$\leq cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3})$$

$$= \frac{16}{13} cn^2 + \Theta(n^{\log_4 3})$$

$$\boxed{\sum_{i=0}^{\infty} r^i = \frac{1}{1-r},\ \text{se } |r| < 1}$$

Série geométrica

Como $n^{\log_4 3} < n^2$, $\frac{16}{13}cn^2$ é "dominante" e $T(n) = O(n^2)$.

(15) ## Método mestre

Recorrências de algoritmos que requerem chamadas recursivas para resolver problemas de tamanho (n/b) com custo $f(n)$ em passos de divisão e conquista podem ser escritas (com constantes $a \geq 1$ e $b > 1$) da seguinte forma:

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

Assumindo que a base indutiva tem um custo constante, temos um método geral para descrever solução de algumas destas recorrências.

(16) A recorrência anterior também inclui os casos $\lceil n/b \rceil$ e $\lfloor n/b \rfloor$.

Teorema mestre:

Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função e $T(n)$ definida para inteiros não negativos como a recorrência

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

Então, $T(n)$ tem os seguintes limites assimtóticos:

(17)

(1) Se $f(n) = O(n^{\log_b a - \varepsilon})$ para alguma constante $\varepsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$

(2) Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \cdot \lg n)$

(3) Se $f(n) = \Omega(n^{\log_b a + \varepsilon})$ para alguma constante $\varepsilon > 0$, e se $a f(n/b) \leqslant c f(n)$ para alguma constante $c < 1$ e todos os n suficientemente grandes, então $T(n) = \Theta(f(n))$.

(18) Exemplos:

(a) $T(n) = 9\,T(n/3) + n$

$T(n) = \Theta(n^2)$ (Caso 1, desde que $f(n) = n = O(n^{\log_3 9 - \varepsilon})$)

(b) $T(n) = T(2n/3) + 1$

$T(n) = \Theta(\lg n)$ (Caso 2, desde que $f(n) = 1 = \Theta(n^{\log_{3/2} 1})$)

(c) $T(n) = 3T(n/4) + n \cdot \lg n$

$T(n) = \Theta(n \lg n)$ (Caso 3, desde que $f(n) = n \cdot \lg n = \Omega(n^{\log_4 3 + \varepsilon})$)

19

Lembre-se que

O teorema mestre não resolve todas as recorrências.

Exemplos:

(a) $T(n) = T(n-1) + n$

(b) $2T(n/2) + n \lg n$

20 Exemplo (b):

$$\left.\begin{array}{l} a=2 \\ b=2 \\ f(n) = n \lg n \\ n = n^{\log_b a} = n^{\log_2 2} \end{array}\right\} \text{Parece o caso (3)}$$

$n \cdot \lg n$ é assintoticamente maior que n.

Problema Mas não é polinomialmente maior.

$\frac{n \lg n}{n} = \lg n$ é assintoticamente menor que n^{ε} para qualquer constante $\varepsilon > 0$.