Análise e Síntese de Algoritmos
Divide & Conquer. Teorema Mestre.
CLRS Cap. 4

Prof. Pedro T. Monteiro

IST - Universidade de Lisboa

2024/2025

Contexto

- Revisão [CLRS, Cap.1-13]
  - Fundamentos; notação; exemplos
- Técnicas de Síntese de Algoritmos [CLRS, Cap.15-16]
  - Programação dinâmica [CLRS, Cap.15]
  - Algoritmos greedy [CLRS, Cap.16]
- Algoritmos em Grafos [CLRS, Cap.21-26]
  - Algoritmos elementares
  - Caminhos mais curtos [CLRS, Cap.22,24-25]
  - Árvores abrangentes [CLRS, Cap.23]
  - Fluxos máximos [CLRS, Cap.26]
- Programação Linear [CLRS, Cap.29]
  - Algoritmos e modelação de problemas com restrições lineares
- Tópicos Adicionais
  - Emparelhamento de Cadeias de Caracteres [CLRS, Cap.32]
  - Complexidade Computacional [CLRS, Cap.34]

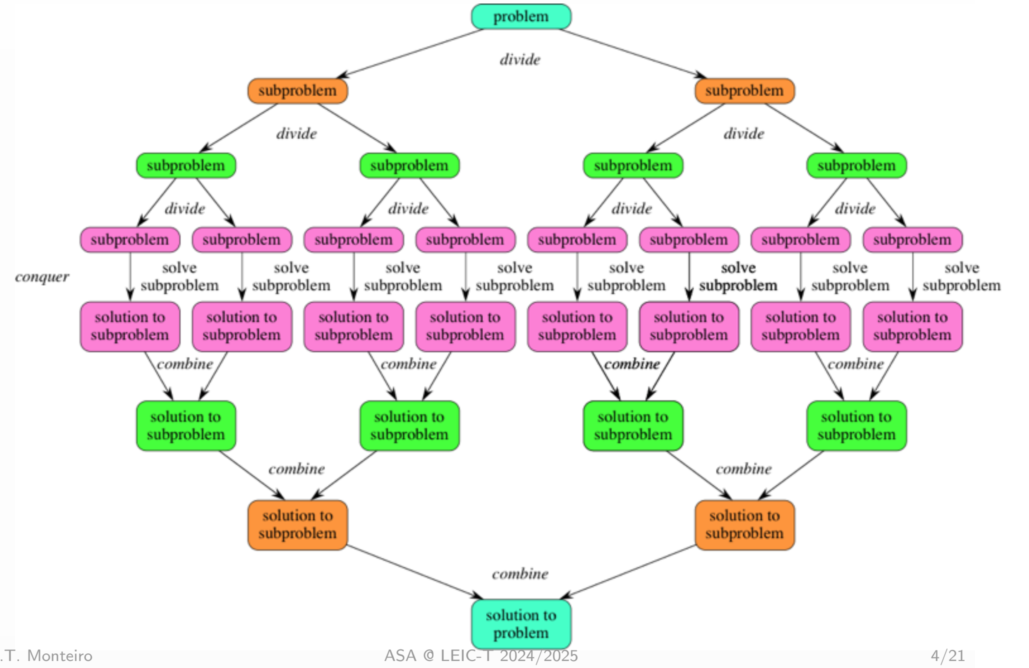
Motivação

Algoritmos - divide & conquer

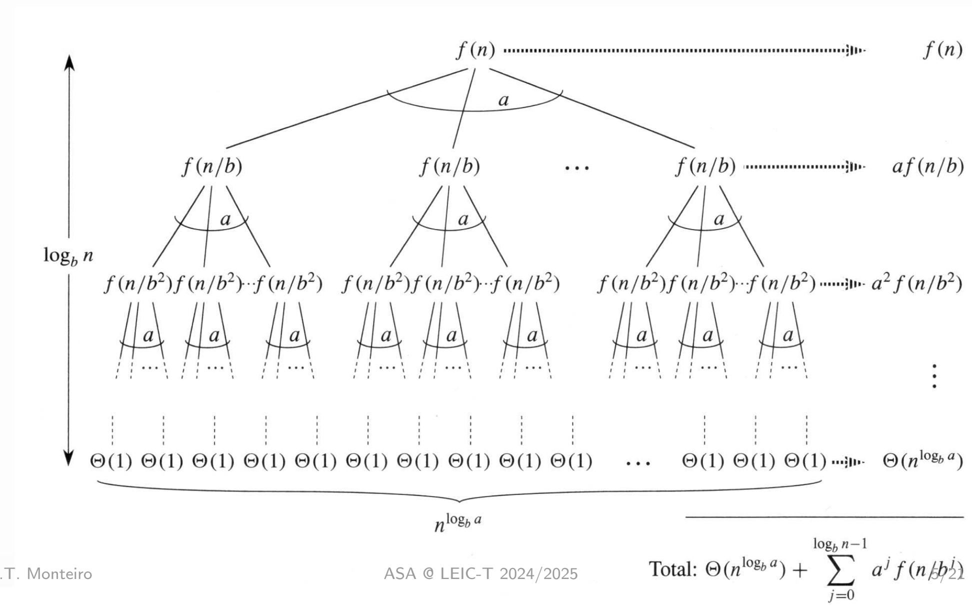
- Partir o problema em sub-problemas (instâncias do mesmo problema)
- Resolver (recursivamente) o sub-problema
- Combinar resultados

Exemplos: mergesort, quicksort, procura binária, min/max, multiplicação matrizes, ...

Motivação



Recorrências divide-and-conquer



Teorema Mestre

Permite resolver recorrências da forma (divide-and-conquer)

$$T(n) = aT(n/b) + \Theta(n^d), \quad a \geq 1, \ b > 1, \ d \geq 0$$

Problema é dividido em a subproblemas, cada um com dimensão n/b

Teorema Mestre

Sejam $a \geq 1, b > 1, d \geq 0$ constantes, e seja $T(n)$ definido por

$$T(n) = aT(n/b) + \Theta(n^d)$$

$T(n)$ é limitado assimptoticamente da seguinte forma:

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } d < \log_b a \\ \Theta(n^d \log n) & \text{if } d = \log_b a \\ \Theta(n^d) & \text{if } d > \log_b a \end{cases}$$

- Em cada um dos 3 casos estamos a comparar d com $\log_b a$
- **Solução da recorrência é determinada pela maior das duas funções!**

Definição do livro Papadimitriou. Definição alternativa no livro do Cormen!

$T(n)$ é limitado assimptoticamente da seguinte forma:

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } d < \log_b a \\ \Theta(n^d \log n) & \text{if } d = \log_b a \\ \Theta(n^d) & \text{if } d > \log_b a \end{cases}$$

Exemplos

$T(n) = 9T(n/3) + n$

- $a = 9, b = 3, d = 1$
- $d = 1$ é $<$ que $\log_3 9$
- $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$ (caso 1 do Teorema Mestre)

Teorema Mestre

$T(n)$ é limitado assimptoticamente da seguinte forma:

$$T(n) = \begin{cases} \Theta(n^{log_b a}) & if\ d < log_b a \\ \Theta(n^d log\ n) & if\ d = log_b a \\ \Theta(n^d) & if\ d > log_b a \end{cases}$$

Exemplos

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

- $a = 1, b = \frac{3}{2}, d = 0$
- $d = 0$ é $=$ a $log_{\frac{3}{2}} 1$
- $T(n) = \Theta(n^d log\ n) = \Theta(log\ n)$ (caso 2 do Teorema Mestre)

Teorema Mestre

$T(n)$ é limitado assimptoticamente da seguinte forma:

$$T(n) = \begin{cases} \Theta(n^{log_b a}) & if\ d < log_b a \\ \Theta(n^d log\ n) & if\ d = log_b a \\ \Theta(n^d) & if\ d > log_b a \end{cases}$$

Exemplos

$T(n) = 3T(n/3) + n^2$

- $a = 3, b = 3, d = 2$
- $d = 2$ é $>$ que $log_3 3$
- $T(n) = \Theta(n^d) = \Theta(n^2)$ (caso 3 do Teorema Mestre)

Teorema Mestre

Exercício: (casa / quadro)

$T(n) = 4T(n/2) + n^2\sqrt{n}$

- $a = 4, b = 2, d = \frac{5}{2}$
- $d = \frac{5}{2}$ é $>$ que $log_2 4$
- $T(n) = \Theta(n^d) = \Theta(n^{\frac{5}{2}})$ (caso 3 do Teorema Mestre)

Exemplo 1: Mergesort

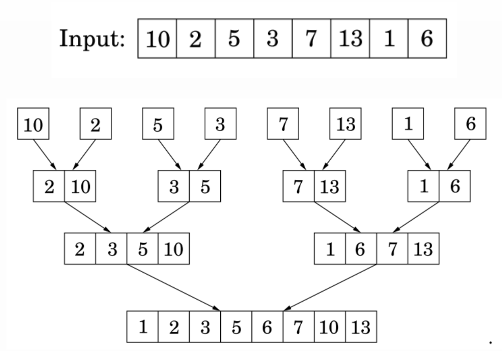
MergeSort(A, p, r)

```
if p < r then
    q ← ⌊(p + r)/2⌋
    MergeSort(A, p, q)
    MergeSort(A, q + 1, r)
    Merge(A, p, q, r)
end if
```

Qual o tempo execução no pior caso?

- Admitindo que tempo de execução de Merge cresce com n

Exemplo 1: Mergesort



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

- $a = 2, b = 2, d = 1$
- $d = 1$ é $=$ a $log_2 2$
- $T(n) = \Theta(n^d \ log \ n) = \Theta(n \ log \ n)$ (caso 2 do Teorema Mestre)

Exemplo 2: Quicksort

partition(A, p, r)

```
x ← A[r]
i ← p - 1
for j ← p to r - 1 do
  if A[j] ≤ x then
    i ← i + 1
    A[i] ↔ A[j]
  end if
end for
A[i+1] ↔ A[j+1]
return i+1
```

Complexidade

- $O(n)$

Propriedades

- todos os elementos à esquerda do pivot são $\leq$ pivot
- todos os elementos à direita do pivot são $>$ pivot
- após execução do partition() o pivot está na sua posição final

Prova por invariante de ciclo

No início de cada iteração do for:

- todos os elementos à esquerda do pivot são $\leq$ pivot
- todos os elementos à direita do pivot são $>$ pivot

Exemplo 2: Quicksort

quicksort(A, p, r)

```
if p < r then
  q ← partition(A, p, r)
  quicksort(A, p, q-1)
  quicksort(A, q+1, r)
end if
```

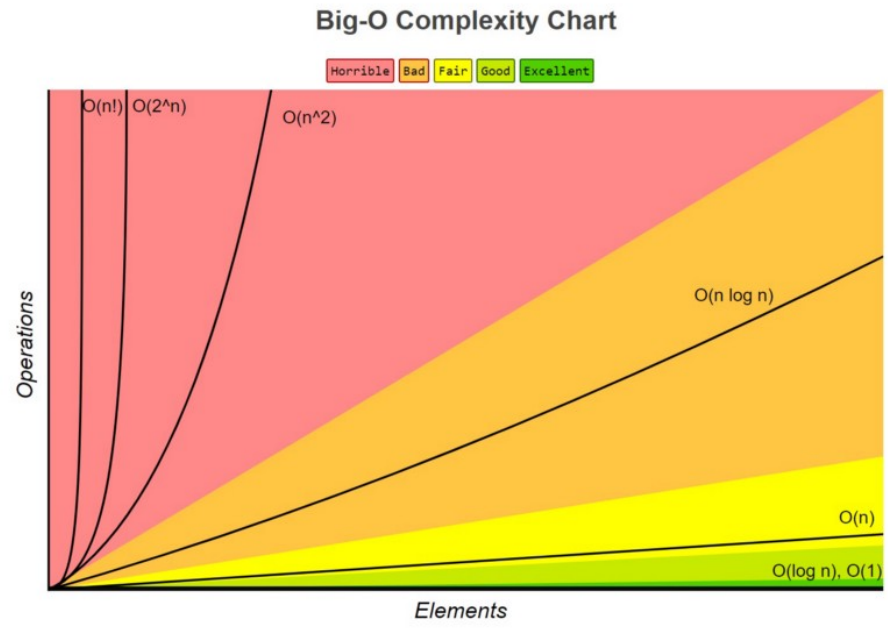
- Vector não necessariamente dividido em 2 partes iguais
- Constantes menores (*in place*)

Complexidade

- Pior caso: $O(n^2)$
- Melhor caso: $O(n \ log \ n)$

Na prática, QuickSort (aleatorizado) é mais rápido que MergeSort

Notação Assimptótica



Exemplo 3: Multiplicação de matrizes

TÉCNICO LISBOA

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

Matrix-multiplication(A, B)

```
n = lines(A)
for i=1 to n do
   for j ← 1 to n do
      c_ij = 0
      for k = 1 to n do
         c_ij ← c_ij + a_ik b_kj
      end for
   end for
end for
```

Complexidade: $\Theta(n^3)$

Exemplo 3: Multiplicação de matrizes

TÉCNICO LISBOA

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

$T(n) = 8T(n/2) + \Theta(n^2)$

- $a = 8, b = 2, d = 2$
- $d = 2$ is $<$ than $log_2 8$
- $T(n) = \Theta(n^{log_b a}) = \Theta(n^3)$ (caso 1 do Teorema Mestre)

Exemplo 3: Multiplicação de matrizes

TÉCNICO LISBOA

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

Algoritmo de Strassen

$M_1 = A(F - H)$
$M_2 = H(A + B)$
$M_3 = E(C + D)$
$M_4 = D(G - E)$
$M_5 = (A + D)(E + H)$
$M_6 = (B - D)(G + H)$
$M_7 = (A - C)(E + F)$

$AE + BG = M_5 + M_4 - M_2 + M_6$
$AF + BH = M_1 + M_2$
$CE + DG = M_3 + M_4$
$CF + DH = M_5 + M_1 - M_3 - M_7$

$T(n) = 7T(n/2) + \Theta(n^2)$

- $a = 7, b = 2, d = 2$
- $d = 2$ is $<$ than $log_2 7$
- $T(n) = \Theta(n^{log_b a}) = \Theta(n^{2.81})$ (caso 1 do Teorema Mestre)

Recorrências

Divide-and-conquer

TÉCNICO LISBOA

Exercício (R1 13/14 II.a): (casa / quadro)

```
int f(int n)
{
  int j, i;

  j = 0;
  i = 0;
  while(i < n)
  {
    j++;
    i+= 2;
  }

  if(n > 1)
    i = 2*f(j) + f(j);

  return i;
}
```

Indique a expressão (recursiva) que descreve o tempo de execução da função em termos do número n, e de seguida, utilizando os métodos que conhece, determine o menor majorante assimptótico.

Questões?

Dúvidas?