Análise de Algoritmos

CLRS 7

Essas transparências foram adaptadas das transparências do Prof. Paulo Feofiloff, do Prof. José Coelho de Pina, e da Profa. Cristina G. Fernandes

Partição

"Problema": Rearranjar um dado vetor $A[p \mathinner{..} r]$ e devolver um índice q tal que $p \le q \le r$ e

$$A[p \mathinner{..} q-1] \le A[q] < A[q+1 \mathinner{..} r]$$

Entra:

	p									r
A	99	33	55	77	11	22	88	66	33	44

Partição

"Problema": Rearranjar um dado vetor $A[p\,..\,r]$ e devolver um índice q tal que $p \leq q \leq r$ e

$$A[p\,..\,q-1] \leq A[q] < A[q+1\,..\,r]$$

Entra:

	p									r
A	99	33	55	77	11	22	88	66	33	44

Possível saída:

	p				q					r
A	33	11	22	33	44	55	99	66	77	88

Partição

"Problema": Rearranjar um dado vetor $A[p\mathinner{..}r]$ e devolver um índice q tal que $p \le q \le r$ e

$$A[p\mathinner{..}q-1] \le A[q] < A[q+1\mathinner{..}r]$$

Entra:

		p									r
A		99	33	55	77	11	22	88	66	33	44

Outra possível saída:

		p				q					r
A		33	22	33	11	44	77	55	99	66	88

Partição

"Problema": Rearranjar um dado vetor $A[p \mathinner{..} r]$ e devolver um índice q tal que $p \leq q \leq r$ e

$$A[p \mathinner{..} q-1] \leq A[q] < A[q+1 \mathinner{..} r]$$

Entra:

	p									r
A	99	33	55	77	11	22	88	66	33	44

Mais uma possível saída:

	p						q			r
A	33	22	33	11	44	55	66	99	77	88

Partição

"Problema": Rearranjar um dado vetor $A[p\,.\,.\,r]$ e devolver um índice q tal que $p \leq q \leq r$ e

$$\underbrace{A[p\,.\,.\,q-1]}_{A[p\,.\,.\,q)} \leq A[q] < \underbrace{A[q+1\,.\,.\,r]}_{A(q\,.\,.\,r]}$$

Entra:

	p									r
A	99	33	55	77	11	22	88	66	33	44

Mais uma possível saída:

	p					q				r
A	33	22	33	11	44	55	66	99	77	88

Partição

"Problema": Rearranjar um dado vetor $A[p \mathinner{.\,.} r)$ e devolver um índice q tal que $p \le q < r$ e

$$\underbrace{A[p \mathinner{.\,.} q-1]}_{A[p \mathinner{.\,.} q)} \le A[q] < \underbrace{A[q+1 \mathinner{.\,.} r-1]}_{A(q \mathinner{.\,.} r)}$$

Entra:

	p										r
A	99	33	55	77	11	22	88	66	33	44	

Mais uma possível saída:

	p					q					r
A	33	22	33	11	44	55	66	99	77	88	

Estamos em boa companhia: Dijkstra

EWD831 - 0

Why numbering should start at zero **EWD831.html**

To denote the subsequence of natural numbers $2, 3, \ldots, 12$ without the pernicious three dots, four conventions are open to us:

a) $2 \le i < 13$

b) $1 < i \le 12$

c) $2 \le i \le 12$

d) $1 < i < 13$

Partição

"Problema": Rearranjar um dado vetor $A[p \mathinner{.\,.} r)$ e devolver um índice q tal que $p \leq q < r$ e

$$A[p \mathinner{.\,.} q) \leq A[q] < A(q \mathinner{.\,.} r)$$

Entra:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} & p & & & & & & & & & & r \\ \hline A & 99 & 33 & 55 & 77 & 11 & 22 & 88 & 66 & 33 & 44 & \\ \hline \end{array}$$

Primeira chamada durante Quicksort: PARTICIONE $(A, 0, n)$

Aqui estamos supondo $A[0 \mathinner{.\,.} n-1]$, ou seja, $A[0 \mathinner{.\,.} n)$.

Particione

	p										r
A	99	33	55	77	11	22	88	66	33	44	

Particione

	$i = j$								x	r
A	99	33	55	77	11	22	88	66	33	44	

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

		i		j						x
A	33	99	55	77	11	22	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

		i			j					x
A	33	99	55	77	11	22	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

			i			j				x
A	33	11	55	77	99	22	88	66	33	44

Invariantes:

(i0) $A[p\,.\,.\,i) \leq x$ (i1) $x < A[i\,.\,.\,j)$ (i2) $A[r-1] = x$

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

			i			j				x
A	33	11	55	77	99	22	88	66	33	44

				i			j			x
A	33	11	22	77	99	55	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

			i			j				x
A	33	11	55	77	99	22	88	66	33	44

				i			j			x
A	33	11	22	77	99	55	88	66	33	44

				i				j		x
A	33	11	22	77	99	55	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

			i			j				x
A	33	11	55	77	99	22	88	66	33	44

				i			j			x
A	33	11	22	77	99	55	88	66	33	44

				i					j	x
A	33	11	22	77	99	55	88	66	33	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

			i			j				x
A	33	11	55	77	99	22	88	66	33	44

				i			j			x
A	33	11	22	77	99	55	88	66	33	44

					i					j
A	33	11	22	33	99	55	88	66	77	44

Particione

	i	j								x
A	99	33	55	77	11	22	88	66	33	44

		i	j							x
A	33	99	55	77	11	22	88	66	33	44

			i			j				x
A	33	11	55	77	99	22	88	66	33	44

				i			j			x
A	33	11	22	77	99	55	88	66	33	44

					i					j
A	33	11	22	33	99	55	88	66	77	44

	p				q						r
A	33	11	22	33	44	55	88	66	77	99	

Particione

Rearranja $A[p\,..\,r)$ de modo que $p \leq q < r$ e
$A[p\,..\,q) \leq A[q] < A(q\,..\,r)$

```
PARTICIONE (A, p, r)
1  x ← A[r−1]                ▷ x é o "pivô"
2  i ← p
3  para j ← p até r − 2 faça
4      se A[j] ≤ x
5          então A[i] ↔ A[j]
6                i ← i + 1
7  A[i] ↔ A[r − 1]           ▷ troca com o "pivô"
8  devolva i
```

Invariantes: no começo de cada iteração de 3–6,

(i0) $A[p\,..\,i) \leq x$ (i1) $x < A[i\,..\,j)$ (i2) $A[r-1] = x$

Consumo de tempo

Quanto tempo consome em função de $n := r - p$?

linha	consumo de todas as execuções da linha
1-2	$= 2\,\Theta(1)$
3	$= \Theta(n)$
4	$= \Theta(n)$
5-6	$= 2\,\mathrm{O}(n)$
7-8	$= 2\,\Theta(1)$
total	$= 2 \cdot \Theta(n) + 4 \cdot \Theta(1) + 2 \cdot \mathrm{O}(n) = \Theta(n)$

Conclusão:

> O algoritmo PARTICIONE consome tempo $\Theta(n)$.

Quicksort

Rearranja $A[p\,..\,r)$ em ordem crescente.

QUICKSORT (A, p, r)
se $r - p > 1$
  então $q \leftarrow$ PARTICIONE(A, p, r)
    QUICKSORT(A, p, q)
    QUICKSORT$(A, q+1, r)$

	p										r
A	99	33	55	77	11	22	88	66	33	44	

Quicksort

Rearranja $A[p\,..\,r)$ em ordem crescente.

QUICKSORT (A, p, r)
1 **se** $p < r$
2 **então** $q \leftarrow$ PARTICIONE (A, p, r)
3 QUICKSORT (A, p, q)
4 QUICKSORT $(A, q+1, r)$

	p				q						r
A	33	11	22	33	44	55	88	66	77	99	

No começo da linha 3,

$$A[p\,..\,q) \leq A[q] < A[q{+}1\,..\,r)$$

Quicksort

Rearranja $A[p\,..\,r)$ em ordem crescente.

QUICKSORT (A, p, r)
se $p < r$
  então $q \leftarrow$ PARTICIONE (A, p, r)
    QUICKSORT (A, p, q)
    QUICKSORT $(A, q+1, r)$

	p				q						r
A	11	22	33	33	44	55	88	66	77	99	

Quicksort

Rearranja $A[p \,..\, r)$ em ordem crescente.

QUICKSORT (A, p, r)
se $p < r$
  então $q \leftarrow$ PARTICIONE (A, p, r)
    QUICKSORT (A, p, q)
    QUICKSORT $(A, q + 1, r)$

	p				q						r
A	11	22	33	33	44	55	66	77	88	99	

Quicksort

Rearranja $A[p \mathinner{..} r)$ em ordem crescente.

QUICKSORT (A, p, r)
1 **se** $p < r$
2 **então** $q \leftarrow$ PARTICIONE (A, p, r)
3 QUICKSORT (A, p, q)
4 QUICKSORT $(A, q + 1, r)$

No começo da linha 3,

$$A[p \mathinner{..} q) \leq A[q] < A[q{+}1 \mathinner{..} r)$$

Consumo de tempo?

Quicksort

Rearranja $A[p \,.\,.\, r)$ em ordem crescente.

```
QUICKSORT (A, p, r)
1  se p < r
2     então q ← PARTICIONE (A, p, r)
3           QUICKSORT (A, p, q)
4           QUICKSORT (A, q + 1, r)
```

No começo da linha 3,

$$A[p \,.\,.\, q) \leq A[q] < A[q{+}1 \,.\,.\, r)$$

Consumo de tempo?

$T(n) :=$ consumo de tempo no pior caso sendo $n := r - p$

Consumo de tempo

Quanto tempo consome em função de $n := r - p$?

linha		consumo de tempo máximo
1	$=$	?
2	$=$	?
3	$=$	?
4	$=$	?
total	$=$	????

Consumo de tempo

Quanto tempo consome em função de $n := r - p$?

linha		consumo de tempo máximo
1	$=$	$\Theta(1)$
2	$=$	$\Theta(n)$
3	$\leq$	$T(k)$
4	$\leq$	$T(n-k-1)$
total	$\leq$	$T(k) + T(n-k-1) + \Theta(n) + \Theta(1)$

para algum $0 \leq k := q - p \leq n - 1$

Recorrência

$T(n) :=$ consumo de tempo quando $n = r - p$

$$T(n) \leq T(k) + T(n-k-1) + \Theta(n)$$

para algum $k \in [0\,..\,n)$.

Recorrência

$T(n) :=$ consumo de tempo máximo quando $n = r - p$

$$T(n) = T(k) + T(n - k - 1) + \Theta(n)$$

para algum $k \in [0 \,..\, n)$.

Recorrência

$T(n)$:= consumo de tempo máximo quando $n = r - p$

$$T(n) = T(k) + T(n - k - 1) + \Theta(n)$$

para algum $k \in [0 \mathinner{..} n)$.

E se $k = 0$ em todas as chamadas recursivas? Pode acontecer?

$$T(n) = T(0) + T(n - 1) + \Theta(n)$$

$T(n)$ é $\Omega(???)$.

Recorrência

$T(n) :=$ consumo de tempo máximo quando $n = r - p$

$$T(n) = T(k) + T(n - k - 1) + \Theta(n)$$

para algum $k \in [0 \,.\,.\, n)$.

E se $k = 0$ em todas as chamadas recursivas? Pode acontecer?

$$T(n) = T(0) + T(n - 1) + \Theta(n)$$

$T(n)$ é $\Omega(n^2)$.

Demonstração: ... Exercício!

Recorrência

$T(n) :=$ consumo de tempo máximo quando $n = r - p$

$$T(n) = \max_{k \in [0\,..\,n)} \Big(T(k) + T(n - k - 1) \Big) + \Theta(n)$$

Recorrência

$T(n)$:= consumo de tempo máximo quando $n = r - p$

$$T(n) = \max_{k \in [0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + \Theta(n)$$

Versão simplificada:

$$T(0) = 1$$
$$T(1) = 1$$
$$T(n) = \max_{k \in [0\,..\,n)} \{ T(k) + T(n-k-1) \} + n \quad \text{para cada } n \in [2\,..\,\infty)$$

n	0	1	2	3	4	5
$T(n)$	1	1	$2+2$	$5+3$	$9+4$	$14+5$

Recorrência

$T(n) :=$ consumo de tempo máximo quando $n = r - p$

$$T(n) = \max_{k \in [0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + \Theta(n)$$

Versão simplificada:

$$T(0) = 1$$
$$T(1) = 1$$
$$T(n) = \max_{k \in [0\,..\,n)} \{ T(k) + T(n-k-1) \} + n \quad \text{para cada } n \in [2\,..\,\infty)$$

n	0	1	2	3	4	5
$T(n)$	1	1	$2+2$	$5+3$	$9+4$	$14+5$

Vamos mostrar que $T(n) \leq n^2 + 1$ para todo $n \geq 0$.

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$\begin{aligned} T(n) &= \max_{k \in [0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\ &\overset{\text{hi}}{\leq} \max_{k \in [0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \end{aligned}$$

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$
\begin{aligned}
T(n) &= \max_{k\in[0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\
&\overset{\text{hi}}{\leq} \max_{k\in[0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \\
&\leq \max_{x\in[0,n-1]} \Big(x^2 + 1 + (n-x-1)^2 + 1 \Big) + n
\end{aligned}
$$

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$
\begin{aligned}
T(n) &= \max_{k\in[0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\
&\overset{\text{hi}}{\leq} \max_{k\in[0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \\
&\leq \max_{x\in[0,n-1]} \Big(x^2 + 1 + (n-x-1)^2 + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(x^2 + 1 + n^2 + x^2 + 1 - 2nx - 2n + 2x + 1 \Big) + n
\end{aligned}
$$

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$
\begin{aligned}
T(n) &= \max_{k\in[0\,..\,n)} \Big(T(k) + T(n{-}k{-}1) \Big) + n \\
&\overset{\text{hi}}{\leq} \max_{k\in[0\,..\,n)} \Big(k^2 + 1 + (n{-}k{-}1)^2 + 1 \Big) + n \\
&\leq \max_{x\in[0,n-1]} \Big(x^2 + 1 + (n{-}x{-}1)^2 + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(x^2 + 1 + n^2 + x^2 + 1 - 2nx - 2n + 2x + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(2x^2 - 2x(n-1) + 3 + n^2 - 2n \Big) + n
\end{aligned}
$$

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$
\begin{aligned}
T(n) &= \max_{k\in[0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\
&\overset{\text{hi}}{\leq} \max_{k\in[0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \\
&\leq \max_{x\in[0,n-1]} \Big(x^2 + 1 + (n-x-1)^2 + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(x^2 + 1 + n^2 + x^2 + 1 - 2nx - 2n + 2x + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(2x^2 - 2x(n-1) + 3 + n^2 - 2n \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(2x^2 - 2x(n-1) \Big) + 3 + n^2 - 2n + n
\end{aligned}
$$

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$
\begin{aligned}
T(n) &= \max_{k\in[0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\
&\overset{\text{hi}}{\leq} \max_{k\in[0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \\
&\leq \max_{x\in[0,n-1]} \Big(x^2 + 1 + (n-x-1)^2 + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(2x^2 - 2x(n-1) \Big) + 3 + n^2 - 2n + n \\
&= 2 \max_{x\in[0,n-1]} \Big(x^2 - x(n-1) \Big) + 3 + n^2 - n
\end{aligned}
$$

Demonstração

Prova: Trivial para $n \le 1$. Se $n \ge 2$ então

$$
\begin{aligned}
T(n) &= \max_{k \in [0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\
&\overset{\text{hi}}{\le} \max_{k \in [0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \\
&\le \max_{x \in [0, n-1]} \Big(x^2 + 1 + (n-x-1)^2 + 1 \Big) + n \\
&= \max_{x \in [0, n-1]} \Big(2x^2 - 2x(n-1) \Big) + 3 + n^2 - 2n + n \\
&= 2 \max_{x \in [0, n-1]} \Big(x^2 - x(n-1) \Big) + 3 + n^2 - n \\
&= 2 \max \Big\{ 0, (n-1)^2 - (n-1) \cdot (n-1) \Big\} + 3 + n^2 - n
\end{aligned}
$$

Demonstração

Prova: Trivial para $n \leq 1$. Se $n \geq 2$ então

$$\begin{aligned}
T(n) &= \max_{k\in[0\,..\,n)} \Big(T(k) + T(n-k-1) \Big) + n \\
&\overset{\text{hi}}{\leq} \max_{k\in[0\,..\,n)} \Big(k^2 + 1 + (n-k-1)^2 + 1 \Big) + n \\
&\leq \max_{x\in[0,n-1]} \Big(x^2 + 1 + (n-x-1)^2 + 1 \Big) + n \\
&= \max_{x\in[0,n-1]} \Big(2x^2 - 2x(n-1) \Big) + 3 + n^2 - 2n + n \\
&= 2 \max_{x\in[0,n-1]} \Big(x^2 - x(n-1) \Big) + 3 + n^2 - n \\
&= 2 \max \Big\{ 0, (n-1)^2 - (n-1)\cdot(n-1) \Big\} + 3 + n^2 - n \\
&= n^2 - n + 3 \leq n^2 + 1.
\end{aligned}$$

Algumas conclusões

$T(n)$ é $\Theta(n^2)$.

O consumo de tempo do QUICKSORT no pior caso é $\Theta(n^2)$.

O consumo de tempo do QUICKSORT é $\mathrm{O}(n^2)$.

Análise de caso médio do Quicksort

Apesar de o consumo de tempo de pior caso do QUICKSORT ser $\Theta(n^2)$, sua performance na prática é comparável (e em geral melhor) a de outros algoritmos cujo consumo de tempo no pior caso é $\mathrm{O}(n \lg n)$.

Por que isso acontece?

Exercício

Considere a recorrência

$$T(1) = 1$$
$$T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n$$

para $n = 2, 3, 4, \ldots$

Solução assintótica: $T(n)$ é $\mathrm{O}(???)$

Exercício

Considere a recorrência

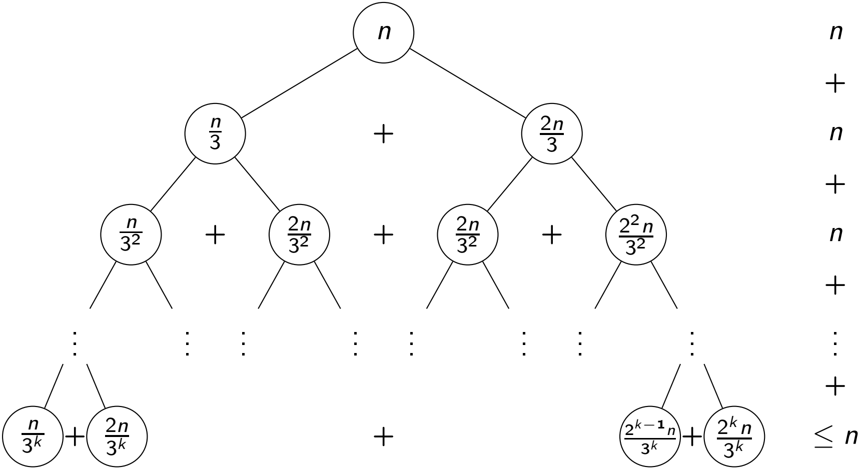
$$T(1) = 1$$
$$T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n$$

para $n = 2, 3, 4, \ldots$

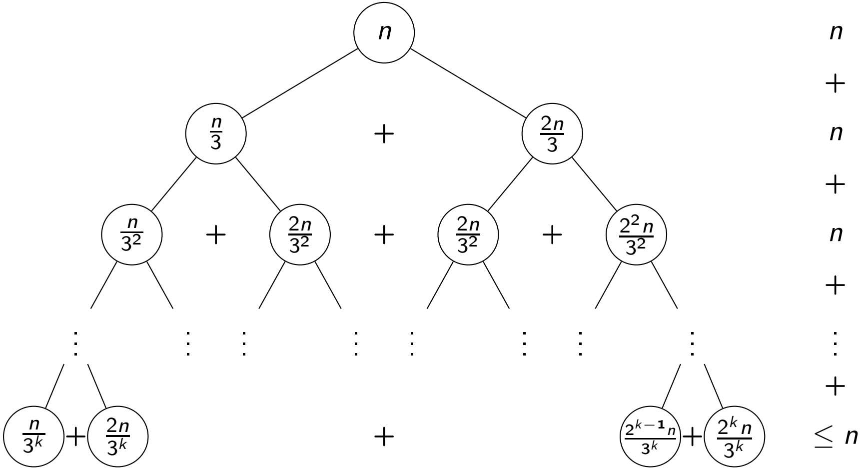
Solução assintótica: $T(n)$ é $\mathrm{O}(???)$

Vamos olhar a árvore da recorrência.

Árvore da recorrência

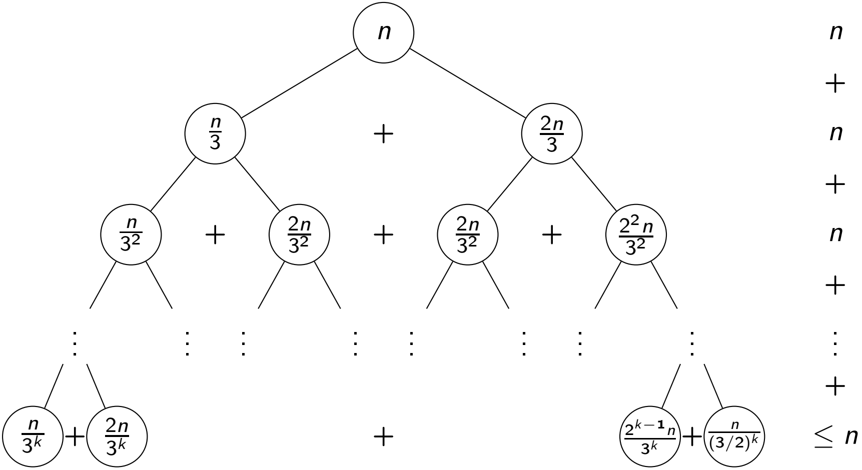


Árvore da recorrência



Os níveis da esquerda chegarão antes na base, ou seja, a árvore será inclinada para a direita.

Árvore da recorrência



Os níveis da esquerda chegarão antes na base, ou seja, a árvore será inclinada para a direita.

Árvore da recorrência

soma em cada horizontal $\leq n$

número de "níveis" $\leq \log_{3/2} n$

$T(n) =$ a soma de tudo

$$T(n) \;\leq\; n \log_{3/2} n \;+\; \underbrace{1 + \cdots + 1}_{\#\text{ de folhas}}$$

Chute: $T(n)$ é $\mathrm{O}(n \lg n)$.

De volta a recorrência

$$T(1) = 1$$

$$T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n \quad \text{para todo } n \in [2\,..\,\infty)$$

n		$T(n)$
1		1
2	$1 + 1 + 2 =$	4
3	$1 + 4 + 3 =$	8
4	$4 + 4 + 4 =$	12

De volta a recorrência

$$T(1) = 1$$

$$T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n \quad \text{para todo } n \in [2 \,..\, \infty)$$

n		$T(n)$
1		1
2	$1 + 1 + 2 =$	4
3	$1 + 4 + 3 =$	8
4	$4 + 4 + 4 =$	12

Queremos: contante C tal que $T(n) \leq C n \lg n$ para todo $n \in [2 \,..\, \infty)$.

De volta a recorrência

$$T(1) = 1$$

$$T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n \quad \text{para todo } n \in [2\,..\,\infty)$$

n		$T(n)$
1		1
2	$1+1+2=$	4
3	$1+4+3=$	8
4	$4+4+4=$	12

Queremos: contante C tal que $T(n) \leq C\,n \lg n$ para todo $n \in [2\,..\,\infty)$.

Para $n = 2$ temos $T(2) = 4 \leq C \cdot 2 \cdot \lg 2$ sse $C \geq 2$.
Para $n = 3$ temos $T(3) = 8 \leq C \cdot 3 \cdot \lg 3$ se $C \geq 3$.

Seja $n \geq 4$. Então...

Rascunho da prova

$$
\begin{aligned}
T(n) &= T(\lceil \tfrac{n}{3} \rceil) + T(\lfloor \tfrac{2n}{3} \rfloor) + n \\
&\overset{\text{hi}}{\leq} C \lceil \tfrac{n}{3} \rceil \lg \lceil \tfrac{n}{3} \rceil + C \lfloor \tfrac{2n}{3} \rfloor \lg \lfloor \tfrac{2n}{3} \rfloor + n \\
&\leq C \tfrac{n+2}{3} \lceil \lg \tfrac{n}{3} \rceil + C \tfrac{2n}{3} \lg \tfrac{2n}{3} + n \\
&< C \tfrac{n+2}{3} (\lg \tfrac{n}{3} + 1) + C \tfrac{2n}{3} \lg \tfrac{2n}{3} + n \\
&= C \tfrac{n+2}{3} \lg \tfrac{2n}{3} + C \tfrac{2n}{3} \lg \tfrac{2n}{3} + n \\
&= C \tfrac{n}{3} \lg \tfrac{2n}{3} + \tfrac{2}{3} C \lg \tfrac{2n}{3} + C \tfrac{2n}{3} \lg \tfrac{2n}{3} + n \\
&= C n \lg \tfrac{2n}{3} + \tfrac{2}{3} C \lg \tfrac{2n}{3} + n
\end{aligned}
$$

Continuação do rascunho da prova

$$
\begin{aligned}
&< \quad Cn\lg\tfrac{2n}{3} + \tfrac{2}{3}C\lg\tfrac{2n}{3} + n \\
&= \quad Cn\lg n + Cn\underbrace{\lg\tfrac{2}{3}}_{\approx -0{,}5849625 < -0{,}5} + \tfrac{2}{3}C\lg n + \tfrac{2}{3}C\lg\tfrac{2}{3} + n \\
&< \quad Cn\lg n + Cn(-\tfrac{1}{2}) + \tfrac{2}{3}C\lg n + \tfrac{2}{3}C(-\tfrac{1}{2}) + n \\
&= \quad Cn\lg n - \tfrac{C}{2}n + \tfrac{2}{3}C\lg n - \tfrac{C}{3} + n \\
&= \quad Cn\lg n - \left(\tfrac{C}{2} - 1\right)n + \tfrac{2}{3}C\lg n - \tfrac{C}{3} \\
&\underbrace{\leq}_{\text{quero}} \quad Cn\lg n \quad \Longleftrightarrow \quad \tfrac{2}{3}C\lg n \leq \left(\tfrac{C}{2} - 1\right)n + \tfrac{C}{3}
\end{aligned}
$$

Escolha da constante

Parece que a prova vai funcionar se escolhermos uma constante $C \geq 3$ tal que, para todo $n \in [4 \,..\, \infty)$, vale que

$$\tfrac{2}{3} C \lg n \leq \left(\tfrac{C}{2} - 1\right) n + \tfrac{C}{3}$$

Escolha da constante

Parece que a prova vai funcionar se escolhermos uma constante $C \geq 3$ tal que, para todo $n \in [4\,..\,\infty)$, vale que

$$\tfrac{2}{3}C \lg n \leq \left(\tfrac{C}{2} - 1\right)n + \tfrac{C}{3}$$

Tome $C := 6$. A inequação desejada vale sse

$$4 \lg n \leq 2n + 2 \qquad \text{para todo } n \geq 4$$

Escolha da constante

Parece que a prova vai funcionar se escolhermos uma constante $C \geq 3$ tal que, para todo $n \in [4\,..\,\infty)$, vale que

$$\frac{2}{3} C \lg n \leq \left(\frac{C}{2} - 1\right) n + \frac{C}{3}$$

Tome $C := 6$. A inequação desejada vale sse

$$4 \lg n \leq 2n + 2 \qquad \text{para todo } n \geq 4$$

Esta inequação pode ser provada utilizando a mesma técnica utilizada para a prova de que $\lg n \leq n$ para todo $n \geq 2$, na Aula 1.

Passando a prova a limpo

ATENÇÃO: Uma prova adequada teria MAIS passos e justificativas do que o que segue: justificativas para cada inequação utilizada.

Passando a prova a limpo

ATENÇÃO: Uma prova adequada teria MAIS passos e justificativas do que o que segue: justificativas para cada inequação utilizada.

Vamos mostrar que $T(n) \leq 6\, n \lg n$ para todo $n \in [2 \,..\, \infty)$.

Passando a prova a limpo

ATENÇÃO: Uma prova adequada teria MAIS passos e justificativas do que o que segue: justificativas para cada inequação utilizada.

Vamos mostrar que $T(n) \leq 6\,n \lg n$ para todo $n \in [2\,..\,\infty)$.

Para $n = 2$ temos $T(2) = 4 \leq 12 = 6 \cdot 2 \cdot \lg 2$.
Para $n = 3$ temos $T(3) = 8 \leq 18 < 6 \cdot 3 \cdot \lg 3$.

Seja $n \geq 4$. Então...

Continuação da prova

$$\begin{aligned}
T(n) &= T(\lceil \tfrac{n}{3} \rceil) + T(\lfloor \tfrac{2n}{3} \rfloor) + n \\
&\overset{\text{hi}}{\leq} 6\lceil \tfrac{n}{3} \rceil \lg \lceil \tfrac{n}{3} \rceil + 6\lfloor \tfrac{2n}{3} \rfloor \lg \lfloor \tfrac{2n}{3} \rfloor + n \\
&\leq 6 \cdot \tfrac{n+2}{3} \lceil \lg \tfrac{n}{3} \rceil + 6 \cdot \tfrac{2n}{3} \lg \tfrac{2n}{3} + n \\
&< 2(n+2)\,(\lg \tfrac{n}{3} + 1) + 4n \lg \tfrac{2n}{3} + n \\
&= 2(n+2) \lg \tfrac{2n}{3} + 4n \lg \tfrac{2n}{3} + n \\
&= 2n \lg \tfrac{2n}{3} + 4 \lg \tfrac{2n}{3} + 4n \lg \tfrac{2n}{3} + n \\
&= 6n \lg \tfrac{2n}{3} + 4 \lg \tfrac{2n}{3} + n
\end{aligned}$$

Continuação da continuação da prova

$$\begin{aligned}
&< \quad 6n \lg \tfrac{2n}{3} + 4 \lg \tfrac{2n}{3} + n \\
&= \quad 6n \lg n + 6n \underbrace{\lg \tfrac{2}{3}}_{\approx -0{,}5849625 < -0{,}5} + 4 \lg n + 4 \lg \tfrac{2}{3} + n \\
&< \quad 6n \lg n + 6n(-\tfrac{1}{2}) + 4 \lg n + 4(-\tfrac{1}{2}) + n \\
&= \quad 6n \lg n - 3n + 4 \lg n - 2 + n \\
&= \quad 6n \lg n - 2n + 4 \lg n - 2 \\
&\underbrace{\leq}_{\text{agora tenho!}} \quad 6n \lg n
\end{aligned}$$

De volta à intuição

Certifique-se que a conclusão seria a mesma qualquer que fosse a proporção fixa que tomássemos. Por exemplo, resolva o seguinte...

De volta à intuição

Certifique-se que a conclusão seria a mesma qualquer que fosse a proporção fixa que tomássemos. Por exemplo, resolva o seguinte...

Exercício: Considere a recorrência

$$T(1) = 1$$
$$T(n) = T(\lceil n/10 \rceil) + T(\lfloor 9n/10 \rfloor) + n$$

para todo $n \in [2 \,..\, \infty)$ e mostre que $T(n)$ é $\mathrm{O}(n \lg n)$.

De volta à intuição

Certifique-se que a conclusão seria a mesma qualquer que fosse a proporção fixa que tomássemos. Por exemplo, resolva o seguinte...

Exercício: Considere a recorrência

$$T(1) = 1$$
$$T(n) = T(\lceil n/10 \rceil) + T(\lfloor 9n/10 \rfloor) + n$$

para todo $n \in [2\,..\,\infty)$ e mostre que $T(n)$ é $\mathrm{O}(n \lg n)$.

Note que, se o QUICKSORT fizer uma "boa" partição a cada, digamos, 5 níveis da recursão, o efeito geral é o mesmo, assintoticamente, que ter feito uma boa partição em todos os níveis.

Aula que vem

Análise de caso médio do QUICKSORT.

Agora vamos revisar um pouco de
conceitos probabilísticos e ver um exemplo.