AULA 12

i-ésimo menor elemento

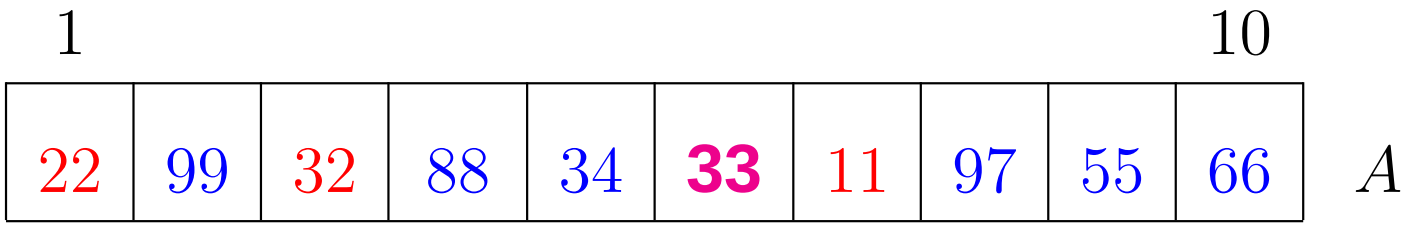
CLRS 9

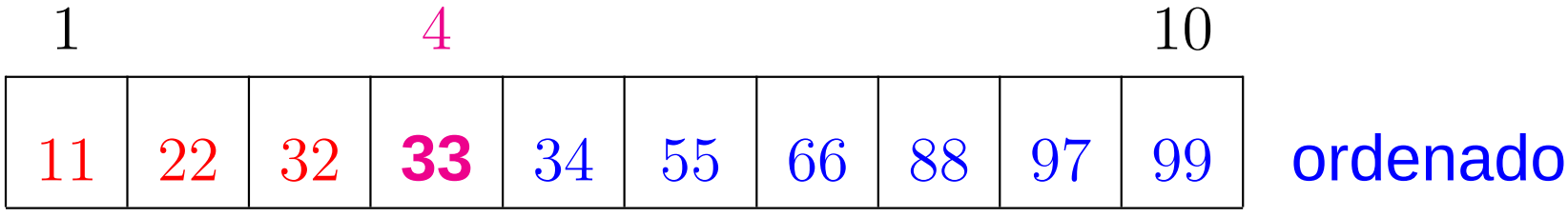
i-ésimo menor

Problema: Encontrar o i-ésimo menor elemento de $A[1\,..\,n]$

Suponha $A[1\,..\,n]$ sem elementos repetidos.

Exemplo: 33 é o 4o. menor elemento de:

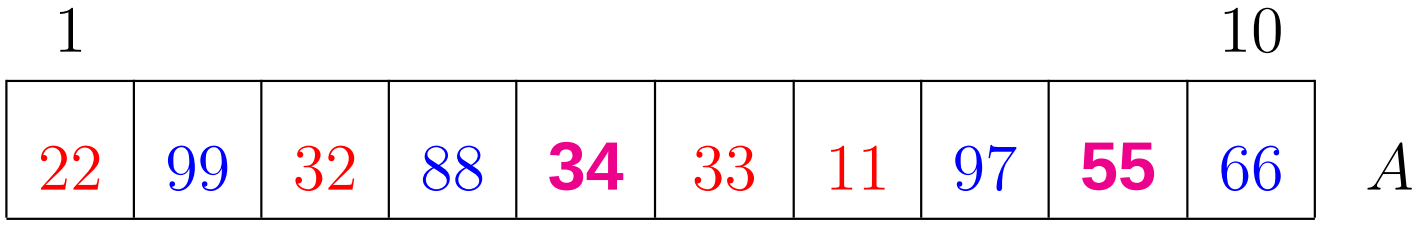


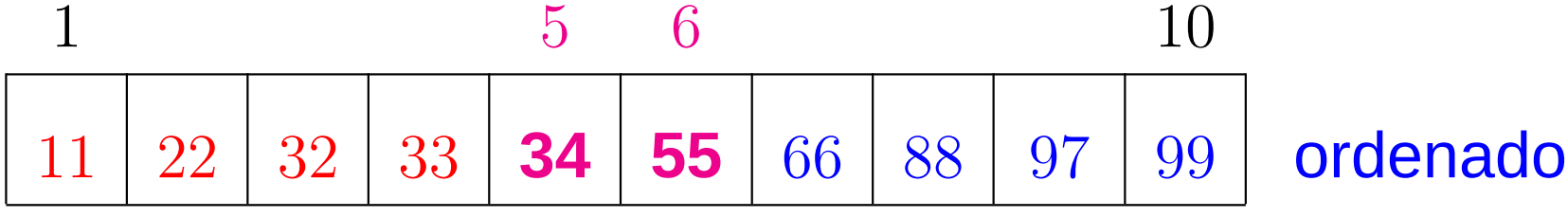


Mediana

Mediana é o $\lfloor \frac{n+1}{2} \rfloor$-ésimo menor ou o $\lceil \frac{n+1}{2} \rceil$-ésimo menor elemento

Exemplo: a mediana é 34 ou 55:





Menor

Recebe um vetor $A[1\,..\,n]$ e devolve o valor do menor elemento.

```
MENOR (A, n)
1    menor ← A[1]
2    para k ← 2 até n faça
3         se A[k] < menor
4              então menor ← A[k]
5    devolva menor
```

O consumo de tempo do algoritmo MENOR é $\Theta(n)$.

Segundo menor

Recebe um vetor $A[1\,..\,n]$ e devolve o valor do segundo menor elemento, supondo $n \geq 2$.

SEG-MENOR (A, n)

```
1   menor ← min{A[1], A[2]}    segmenor ← max{A[1], A[2]}
2   para k ← 3 até n faça
3       se A[k] < menor
4           então segmenor ← menor
5                 menor ← A[k]
6       senão se A[k] < segmenor
7           então segmenor ← A[k]
8   devolva segmenor
```

O consumo de tempo do algoritmo SEG-MENOR é $\Theta(n)$.

i-ésimo menor

Recebe $A[1\,..\,n]$ e i tal que $1 \leq i \leq n$
e devolve valor do i-ésimo menor elemento de $A[1\,..\,n]$

```
SELECT-ORD (A, n, i)
1   ORDENE (A, n)
2   devolva A[i]
```

O consumo de tempo do algoritmo SELECT-ORD é $\Theta(n \lg n)$.

Particione

Rearranja $A[p\,..\,r]$ de modo que $p \le q \le r$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,r]$

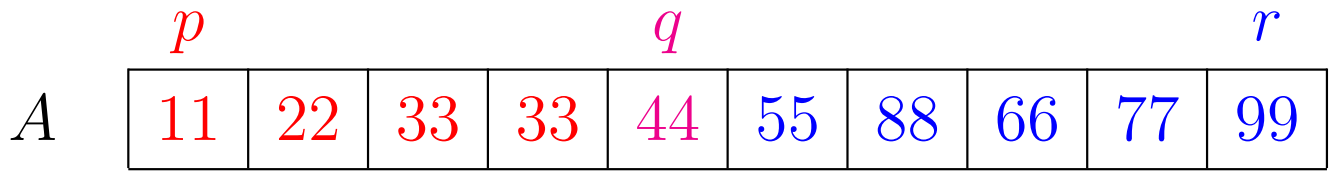
```
PARTICIONE (A, p, r)
1    x ← A[r]          ▷ x é o "pivô"
2    i ← p−1
3    para j ← p até r − 1 faça
4        se A[j] ≤ x
5            então i ← i + 1
6                  A[i] ↔ A[j]
7    A[i+1] ↔ A[r]
8    devolva i + 1
```

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

Particione

Rearranja $A[p\,..\,r]$ de modo que $p \le q \le r$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,r]$

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



Particione

Rearranja $A[p\,..\,r]$ de modo que $p \le q \le r$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,r]$

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

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

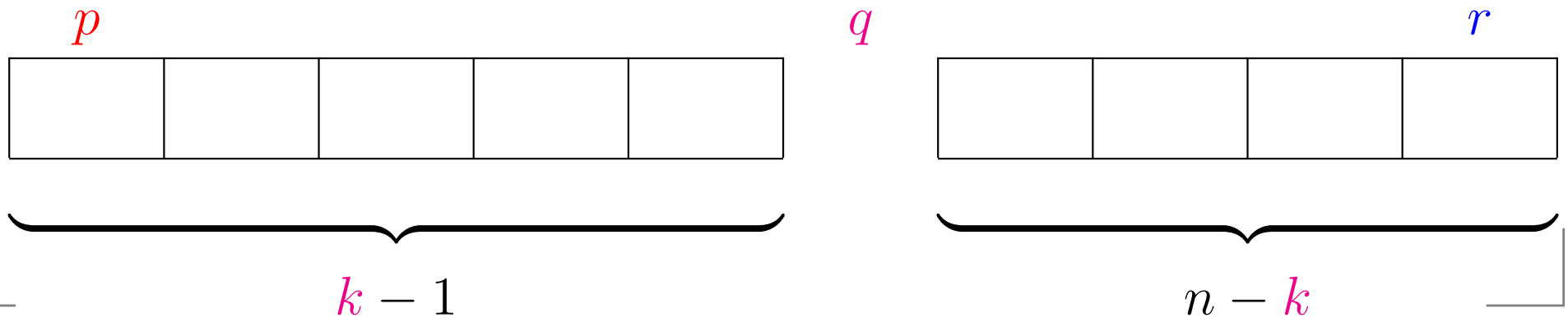
Algoritmo SELECT

Recebe $A[p\,..\,r]$ e i tal que $1 \leq i \leq r-p+1$
e devolve valor do i-ésimo menor elemento de $A[p\,..\,r]$

```
SELECT(A, p, r, i)
1    se p = r
2         então devolva A[p]
3    q ← PARTICIONE (p, r)
4    k ← q − p + 1
5    se k = i
6         então devolva A[q]
7    se k > i
8         então devolva SELECT (A, p, q − 1, i)
9         senão devolva SELECT (A, q + 1, r, i − k)
```

Algoritmo SELECT

```
SELECT(A, p, r, i)
1    se p = r
2         então devolva A[p]
3    q ← PARTICIONE (A, p, r)
4    k ← q − p + 1
5    se k = i
6         então devolva A[q]
7    se k > i
8         então devolva SELECT (A, p, q − 1, i)
9         senão devolva SELECT (A, q + 1, r, i − k)
```



Consumo de tempo

$T(n) =$ consumo de tempo máximo quando $n = r - p + 1$

linha	consumo de todas as execuções da linha
1-2	$= 2\,\Theta(1)$
3	$= \Theta(n)$
4-7	$= 4\,\Theta(1)$
8	$= T(k-1)$
9	$= T(n-k)$

$$T(n) = \Theta(n+6) + \max\{T(k-1), T(n-k)\}$$

$$= \Theta(n) + \max\{T(k-1), T(n-k)\}$$

Consumo de tempo

$T(n)$ pertence a mesma classe Θ que:

$$T'(1) = 1$$
$$T'(n) = T'(n-1) + n \text{ para } n = 2, 3, \ldots$$

Solução assintótica:

$$T'(n) \text{ é } \Theta(n^2).$$

Solução exata:

$$T'(n) = \frac{n^2}{2} + \frac{n}{2}.$$

Algumas conclusões

No melhor caso o consumo de tempo do algoritmo SELECT é $\Theta(n)$.

No pior caso o consumo de tempo do algoritmo SELECT é $\Theta(n^2)$.

Consumo médio?

$\mathrm{E}[T(n)] =$???

Exemplos

Número médio de comparações sobre todas as permutações de $A[p\,..\,r]$ (supondo que nas linhas 8 e 9 o algoritmo sempre escolhe o lado maior):

$A[p\,..\,r]$	comps
1,2	1+0
2,1	1+0
média	2/2

$A[p\,..\,r]$	comps
1,2,3	2+1
2,1,3	2+1
1,3,2	2+0
3,1,2	2+0
2,3,1	2+1
3,2,1	2+1
média	16/6

Mais exemplos

$A[p\,..\,r]$	comps
1,2,3,4	3+3
2,1,3,4	3+3
1,3,2,4	3+2
3,1,2,4	3+2
2,3,1,4	3+3
3,2,1,4	3+3
1,2,4,3	3+1
2,1,4,3	3+1
1,4,2,3	3+1
4,1,2,3	3+1
2,4,1,3	3+1
4,2,1,3	3+1

$A[p\,..\,r]$	comps
1,3,4,2	3+1
3,1,4,2	3+1
1,4,3,2	3+1
4,1,3,2	3+1
3,4,1,2	3+1
4,3,1,2	3+1
2,3,4,1	3+3
3,2,4,1	3+3
2,4,3,1	3+2
4,2,3,1	3+2
3,4,2,1	3+3
4,3,2,1	3+3
média	116/24

Ainda exemplos

No caso $r - p + 1 = 5$, a média é $864/120$.

n	$\mathrm{E}[T(n)]$	
1	0	0
2	$2/2$	1
3	$16/6$	2.7
4	$116/24$	4.8
5	$864/120$	7.2

Número de comparações

O consumo de tempo assintótico é proporcional a

$C(n) =$ número de comparações entre elementos de A quando $n = r - p + 1$

linha	número de comparações na linha
1-2	$= 0$
3	$= n - 1$
4-7	$= 0$
8	$= C(k-1)$
9	$= C(n-k)$
total	$\leq \max\{C(k-1), C(n-k)\} + n - 1$

Número de comparações

No pior caso $C(n)$ pertence a mesma classe Θ que:

$$C'(1) = 0$$

$$C'(n) = C'(n-1) + n - 1 \text{ para } n = 3, 4, \ldots$$

Solução assintótica:

$$C'(n) \text{ é } \Theta(n^2)$$

Solução exata:

$$C'(n) = \frac{n^2}{2} - \frac{n}{2}.$$

Particione aleatorizado

Rearranja $A[p\,..\,r]$ de modo que $p \le q \le r$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,r]$

PARTICIONE-ALEA(A, p, r)
1 $\quad i \leftarrow$ RANDOM (p, r)
2 $\quad A[i] \leftrightarrow A[r]$
3 $\quad$ **devolva** PARTICIONE (A, p, r)

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

SELECT-ALEATORIZADO (= randomized select)

Recebe $A[p\,..\,r]$ e i tal que $1 \le i \le r-p+1$
e devolve valor do i-ésimo menor elemento de $A[p\,..\,r]$

```
SELECT-ALEA(A, p, r, i)
1    se p = r
2         então devolva A[p]
3    q ← PARTICIONE-ALEA (A, p, r)
4    k ← q − p + 1
5    se k = i
6         então devolva A[q]
7    se k > i
8         então devolva SELECT-ALEA (A, p, q − 1, i)
9         senão devolva SELECT-ALEA (A, q + 1, r, i − k)
```

Consumo de tempo

O consumo de tempo é proporcional a

$T(n) =$ número de comparações entre elementos de A quando $n = r - p + 1$

linha	número de comparações na linha
1-2	$= 0$
3	$= n - 1$
4-7	$= 0$
8	$= T(k - 1)$
9	$= T(n - k)$
total	$\leq \max\{T(k-1), T(n-k)\} + n - 1$

$T(n)$ é uma variável aleatória.

Consumo de tempo

$$T(1) = 0$$

$$T(n) \leq \sum_{h=1}^{n-1} X_h T(h) + n - 1 \quad \text{para } n = 2, 3, \ldots$$

onde

$$X_h = \begin{cases} 1 & \text{se } \max\{k-1, n-k\} = h \\ 0 & \text{caso contrário} \end{cases}$$

$$\Pr\{X_h = 1\} = \mathrm{E}[X_h]$$

$$X_h = \begin{cases} 1 & \text{se } \max\{k-1, n-k\} = h \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_h valer 1?

$\Pr\{X_h = 1\} = \mathrm{E}[X_h]$

$$X_h = \begin{cases} 1 & \text{se } \max\{k-1, n-k\} = h \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_h valer 1?

Para $h = 1, \ldots, \lfloor n/2 \rfloor - 1$, $\Pr\{X_h = 1\} = 0 = \mathrm{E}[X_h]$.

Para $h = \lceil n/2 \rceil, \ldots, n$,

$$\Pr\{X_h{=}1\} = \frac{1}{n} + \frac{1}{n} = \frac{2}{n} = \mathrm{E}[X_h]$$

Se n é impar e $h = \lfloor n/2 \rfloor$, então

$$\Pr\{X_h = 1\} = \frac{1}{n} = \mathrm{E}[X_h]$$

Consumo de tempo esperado

$$\mathrm{E}[T(1)] = 0$$

$$\mathrm{E}[T(n)] \leq \sum_{h=1}^{n-1} \mathrm{E}[X_h T(h)] + n - 1$$

$$= \sum_{h=1}^{n-1} \mathrm{E}[X_h]\,\mathrm{E}[T(h)] + n - 1 \quad \text{(CLRS 9.2-2)}$$

$$\leq \frac{2}{n} \sum_{h=a}^{n-1} \mathrm{E}[T(h)] + n - 1 \quad \text{para } n = 2, 3, \ldots$$

onde $a = \lfloor n/2 \rfloor$.

Solução: $\mathrm{E}[T(n)] = \mathrm{O}(n)$.

Consumo de tempo esperado

$\mathrm{E}[T(n)]$ pertence a mesma classe O que:

$$S(1) = 0$$

$$S(n) \leq \frac{2}{n} \sum_{h=a}^{n-1} S(h) + n - 1 \quad \text{para } n = 2, 3, \ldots$$

onde $a = \lfloor n/2 \rfloor$.

n	1	2	3	4	5	6	7	8	9	10
$S(n)$	0.0	1.0	2.7	4.8	7.4	10.0	13.1	15.8	19.4	22.1
$4n$	4	8	12	16	20	24	28	32	36	40

Vamos verificar que $S(n) < 4\,n$ para $n = 1, 2, 3, 4, \ldots$

Recorrência

Prova: Se $n = 1$, então $S(n) = 0 < 4 = 4 \cdot 1 = 4n$. Se $n \geq 2$,

$$S(n) \leq \frac{2}{n} \sum_{h=a}^{n-1} S(h) + n - 1$$

hi

$$< \frac{2}{n} \sum_{h=a}^{n-1} 4\,h + n - 1$$

$$= \frac{8}{n} \left(\sum_{h=1}^{n-1} h - \sum_{h=1}^{a-1} h \right) + n - 1$$

$$\leq \frac{4}{n} \left(n^2 - n - \frac{(n-1)(n-3)}{4} \right) + n - 1$$

$$= \frac{4}{n} \left(\frac{3n^2}{4} - \frac{3}{4} \right) + n - 1$$

$$= 3n - \frac{3}{n} + n - 1 = 4n - \frac{3}{n} - 1 < 4n.$$

Conclusão

O consumo de tempo esperado do algoritmo SELECT-ALEA é $\mathrm{O}(n)$.

Exercícios

Exercício 18.A [CLRS 9.1-1] [muito bom!]
Mostre que o segundo menor elemento de um vetor $A[1\,..\,n]$ pode ser encontrado com não mais que $n + \lceil \lg n \rceil - 2$ comparações.

Exercício 18.B
Prove que o algoritmo Select Aleatorizado (= Randomized Select) funciona corretamente.

Exercício 18.C [CLRS 9.2-3]
Escreva uma versão iterativa do algoritmo Select Aleatorizado (= Randomized Select).

Seleção em tempo linear

CLRS 9.3

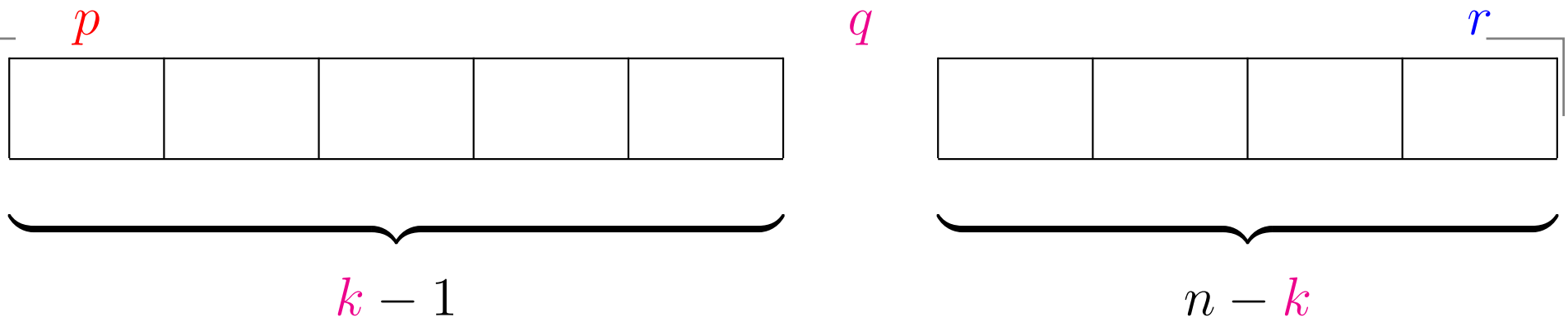
BFPRT = Blum, Floyd, Pratt, Rivest e Tarjan

Select-BFPRT

Recebe $A[p\,..\,r]$ e i tal que $1 \leq i \leq r-p+1$
e devolve um índice q tal que $A[q]$ é o i-ésimo menor elemento de $A[p\,..\,r]$

SELECT-BFPRT(A, p, r, i)
1 **se** $p = r$
2 **então devolva** p ▷ p e não $A[p]$
3 $q \leftarrow$ PARTICIONE-BFPRT (A, p, r)
4 $k \leftarrow q - p + 1$
5 **se** $k = i$
6 **então devolva** q ▷ q e não $A[q]$
7 **se** $k > i$
8 **então devolva** SELECT-BFPRT $(A, p, q-1, i)$
9 **senão devolva** SELECT-BFPRT $(A, q+1, r, i-k)$

Particione-BFPRT



Rearranja $A[p\,..\,r]$ e devolve um índice q, $p \leq q \leq r$, tal que $A[p\,..\,q{-}1] \leq A[q] < A[q{+}1\,..\,r]$ **e**

$$\max\{k-1, n-k\} \leq \left\lfloor \frac{7n}{10} \right\rfloor + 6,$$

onde $n = r - p + 1$ e $k = q - p + 1$.

Suponha que

$P(n) :=$ consumo de tempo máximo do algoritmo PARTICIONE-BFPRT quando $n = r - p + 1$

Consumo de tempo

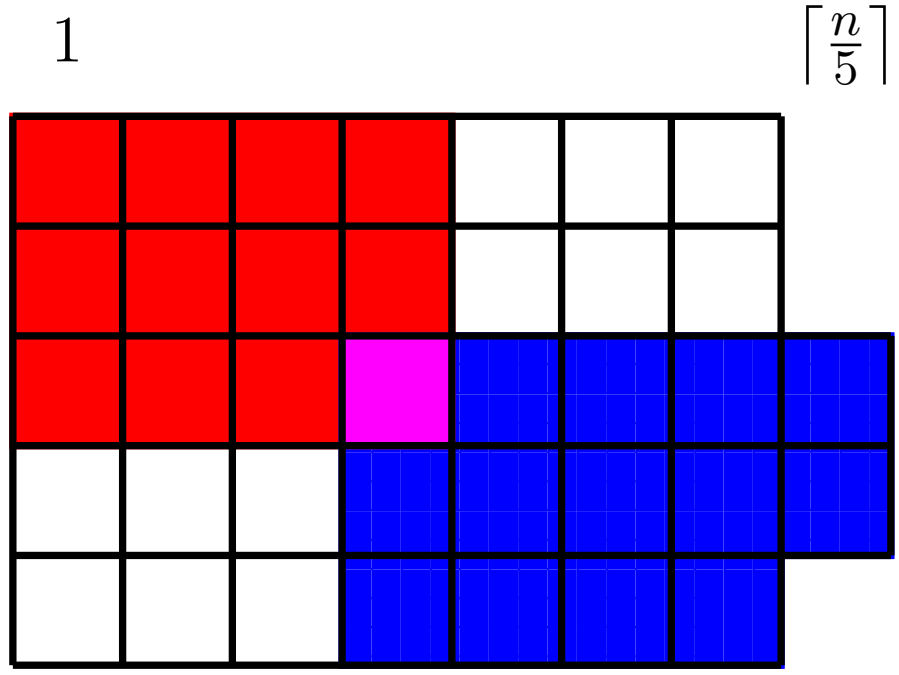
$T(n) :=$ consumo de tempo máximo do algoritmo SELECT-BFPRT quando $n = r - p + 1$

linha	consumo de todas as execuções da linha
1-2	$= 2\,\Theta(1)$
3	$= P(n)$
4-7	$= 4\,\Theta(1)$
8	$= T(k-1)$
9	$= T(n-k)$

$$T(n) = 6\,\Theta(1) + P(n) + \max\{T(k-1), T(n-k)\}$$

$$\leq \Theta(1) + P(n) + T(\lfloor \tfrac{7n}{10} \rfloor + 6)$$

Particione-BFPRT



$$\max\{k-1, n-k\} \leq n - 3\left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5}\right\rceil\right\rceil - 2\right)$$

$$\leq n - \left(\frac{3n}{10} - 6\right) = \frac{7n}{10} + 6$$

Particione-BFPRT

$n := r - p + 1$

PARTICIONE-BFPRT (A, p, r)

1 **para** $j \leftarrow p, p{+}5, p{+}5 \cdot 2, \ldots$ **até** $p{+}5(\lceil n/5 \rceil {-}1)$ **faça**
2 ORDENE $(A, j, j{+}4)$
3 ORDENE $(A, p{+}5\lfloor n/5 \rfloor, n)$

4 **para** $j \leftarrow 1$ **até** $\lceil n/5 \rceil {-}1$ **faça**
5 $B[j] \leftarrow A[p{+}5j{-}3]$
6 $B[\lceil n/5 \rceil] \leftarrow A[\lfloor (p{+}5\lfloor n/5 \rfloor {+}n)/2 \rfloor]$

7 $k \leftarrow$ SELECT-BFPRT$(B, 1, \lceil n/5 \rceil, \lfloor (\lceil n/5 \rceil {+}1)/2 \rfloor)$

8 $A[k] \leftrightarrow A[r]$
9 **devolva** PARTICIONE (A, p, r)

Particione-BFPRT

$n := r - p + 1$

PARTICIONE-BFPRT (A, p, r)

1 **para** $j \leftarrow p, p{+}5, p{+}5 \cdot 2, \ldots$ **até** $p{+}5(\lceil n/5 \rceil {-}1)$ **faça**
2 ORDENE $(A, j, j{+}4)$
3 ORDENE $(A, p{+}5\lfloor n/5 \rfloor, n)$

4 **para** $j \leftarrow 1$ **até** $\lceil n/5 \rceil {-}1$ **faça**
5 $A[j] \leftrightarrow A[p{+}5j{-}3]$
6 $A[\lceil n/5 \rceil] \leftrightarrow A[\lfloor (p{+}5\lfloor n/5 \rfloor {+}n)/2 \rfloor]$

7 $k \leftarrow$ SELECT-BFPRT$(A, p, p{+}\lceil n/5 \rceil {-}1, \lfloor (\lceil n/5 \rceil {+}1)/2 \rfloor)$

8 $A[k] \leftrightarrow A[r]$
9 **devolva** PARTICIONE (A, p, r)

Consumo de tempo do Particione-BFPRT

$P(n) :=$ consumo de tempo máximo do algoritmo PARTICIONE-BFPRT quando $n = r - p + 1$

linha	consumo de todas as execuções da linha
1-3	$= \lceil n/5 \rceil \, \Theta(1)$
4-6	$= \lceil n/5 \rceil \, \Theta(1)$
7	$= T(\lceil n/5 \rceil)$
8	$= \Theta(1)$
9	$= \Theta(n)$

$$P(n) = \Theta(2\lceil n/5 \rceil + n + 1) + T(\lceil n/5 \rceil)$$

$$= \Theta(n) + T(\lceil n/5 \rceil)$$

Consumo de tempo do Select-BFPRT

$T(n) :=$ consumo de tempo máximo do algoritmo SELECT-BFPRT quando $n = r - p + 1$

Temos que

$$T(1) = \Theta(1) \quad \text{para } n < 30$$

$$T(n) \leq \Theta(1) + P(n) + T\left(\left\lfloor \frac{7n}{10} \right\rfloor + 6\right)$$

$$\leq \Theta(1) + \Theta(n) + T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\left\lfloor \frac{7n}{10} \right\rfloor + 6\right)$$

$$= \Theta(n) + T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\left\lfloor \frac{7n}{10} \right\rfloor + 6\right)$$

para $n = 30, 31, \ldots$,

Consumo de tempo do Select-BFPRT

$T(n)$ pertence a mesma classe O que:

$$S(n) = 1 \quad \text{para } n < 30$$

$$S(n) \leq S\left(\left\lceil \frac{n}{5} \right\rceil\right) + S\left(\left\lfloor \frac{7n}{10} \right\rfloor + 6\right) + n \quad \text{para } n \geq 30$$

n	30	60	90	120	150	180	210	240	270	300
$S(n)$	32	185	330	451	572	732	902	1040	1224	1439

Vamos verificar que $S(n) < 80\,n$ para $n = 1, 2, 3, 4, \ldots$

Prova: Se $n = 1, \ldots, 29$, então $S(n) = 1 < 80 < 80n$.
Se $n = 30, \ldots, 99$, então

$$S(n) < S(120) = 451 < 80 \times 30 \leq 80n.$$

Recorrência

Se $n \geq 100$, então

$$S(n) \leq S\left(\left\lceil \frac{n}{5} \right\rceil\right) + S\left(\left\lfloor \frac{7n}{10} \right\rfloor + 6\right) + n$$

$$\overset{\text{hi}}{<} 80\left\lceil \frac{n}{5} \right\rceil + 80\left(\left\lfloor \frac{7n}{10} \right\rfloor + 6\right) + n$$

$$\leq 80\left(\frac{n}{5} + 1\right) + 80\left(\frac{7n}{10} + 6\right) + n$$

$$= 80\frac{n}{5} + 80 + 80\frac{7n}{10} + 480 + n$$

$$= 16n + 56n + n + 560$$

$$= 73n + 560$$

$$< 80n \quad \text{(pois } n \geq 100\text{)}.$$

Logo, $T(n)$ é $O(n)$.

Conclusão

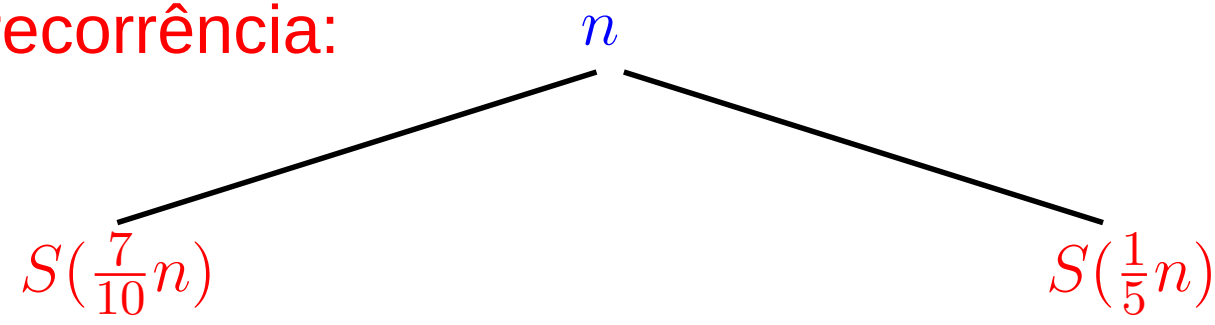
O consumo de tempo do algoritmo SELECT-BFPRT é $\mathrm{O}(n)$.

Como adivinhei classe O?

Árvore da recorrência: $S(n)$

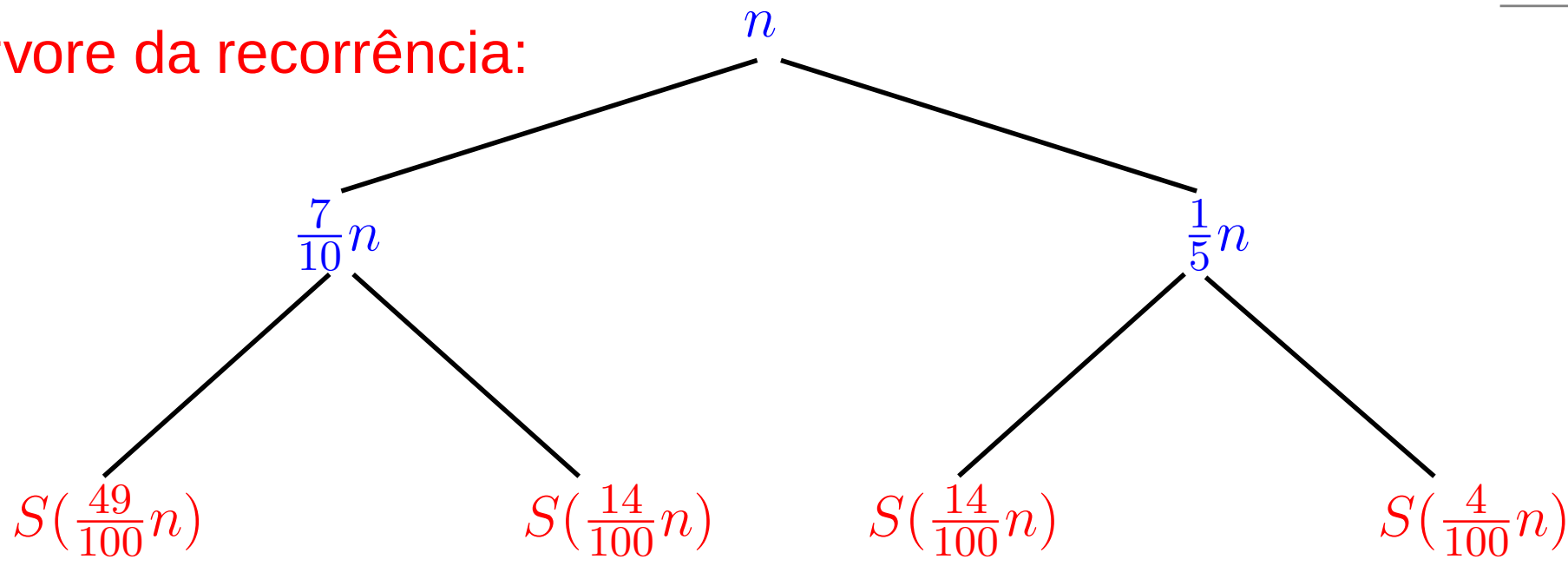
Como adivinhei classe O?

Árvore da recorrência:



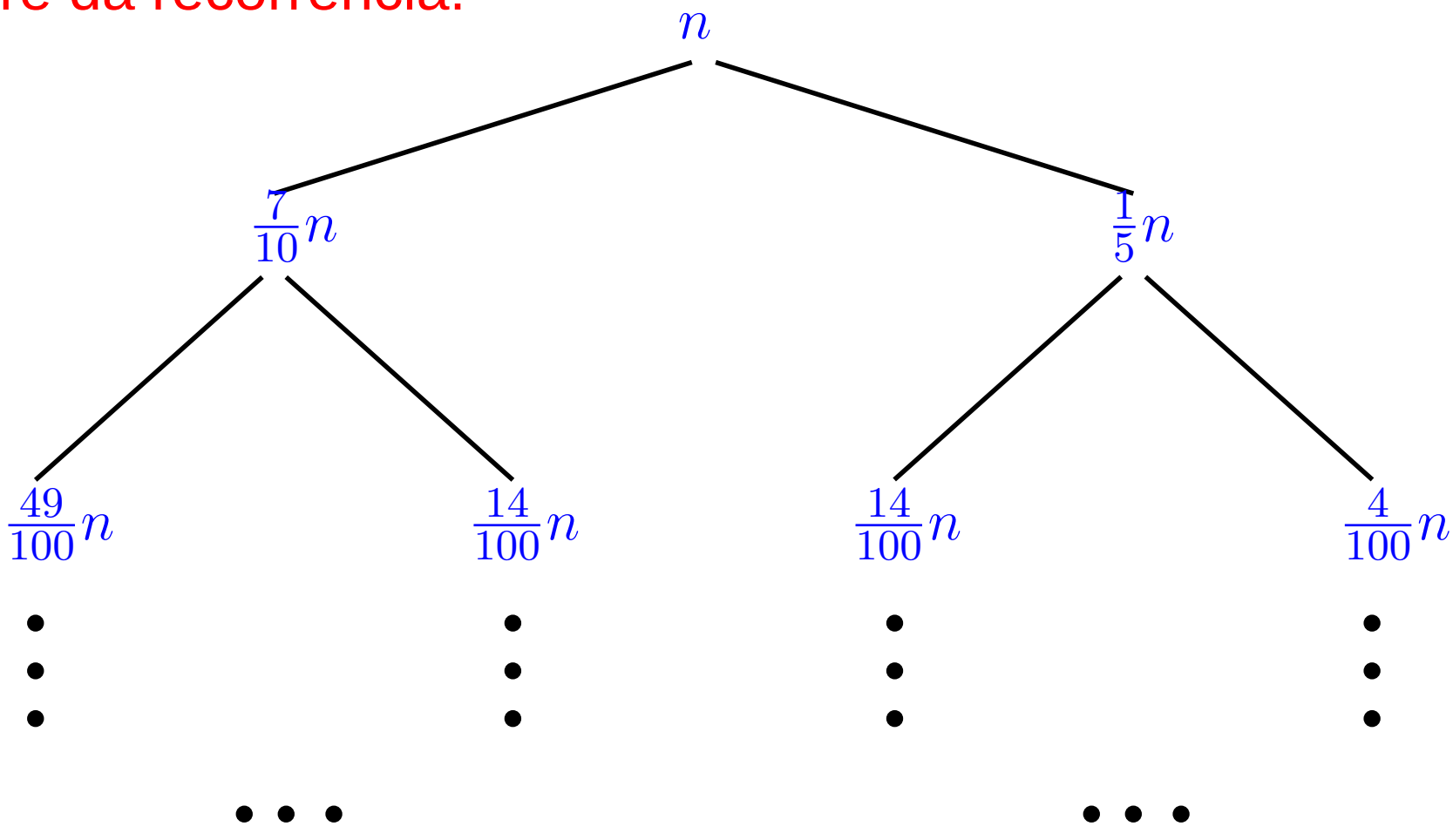
Como adivinhei classe O?

Árvore da recorrência:



Como adivinhei classe O?

Árvore da recorrência:



Contabilidade

nível	0	1	2	$\ldots$	$k-1$	k
soma	n	$\frac{9}{10}n$	$\frac{9^2}{10^2}n$	$\ldots$	$\frac{9^{k-1}}{10^{k-1}}n$	$\frac{9^k}{10^k}n$

$$\frac{10^{k-1}}{9^{k-1}} < n \leq \frac{10^k}{9^k} \Rightarrow k = \lceil \log_{\frac{10}{9}} n \rceil$$

$$\begin{aligned} S(n) &= n + \frac{9}{10}n + \cdots + \frac{9^{k-1}}{10^{k-1}}n + \frac{9^k}{10^k}n \\ &= (1 + \frac{9}{10} + \cdots + \frac{9^k}{10^k})n \\ &= 10(1 - \frac{9^{k+1}}{10^{k+1}})n \\ &< 10n \end{aligned}$$