k-ésimo menor elemento

CLRS Sec 9.3

k-ésimo menor

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

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

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

1									10	
22	99	32	88	34	**33**	11	97	55	66	A

1			4						10	
11	22	32	**33**	34	55	66	88	97	99	ordenado

k-ésimo menor

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

```
SELECT-ORD (A, p, r, k)
1  ORDENE (A, p, r)
2  devolva A[p − 1 + k]
```

SELECT-ORD pode ser implementado de modo a consumir tempo $\mathrm{O}(n \lg n)$ onde $n := r - p$.

k-ésimo menor

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

```
SELECT-ORD (A, p, r, k)
1  ORDENE (A, p, r)
2  devolva A[p − 1 + k]
```

SELECT-ORD pode ser implementado de modo a consumir tempo $\mathrm{O}(n \lg n)$ onde $n := r - p$.

Dá para fazer melhor?

Relembremos o Particione

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

```
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:

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

Relembremos o Particione

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

```
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
```

Consumo de tempo: $\Theta(n)$ onde $n := r - p$.

Algoritmo SELECT determinístico

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

```
SELECT (A, p, r, k)     ▷ devolve k-ésimo menor de A[p .. r)
1   se r − p = 1 então devolva A[p]
2   q ← PARTICIONE (A, p, r)
3   num≤pivô ← q − p + 1     ▷ nº de elementos em A[p .. q]
4   se k = num≤pivô
5       então devolva A[q]
6   se k < num≤pivô
7       então devolva SELECT (A, p, q, k)
8       senão devolva SELECT (A, q+1, r, k−num≤pivô)
```

Consumo de tempo no pior caso é $\Theta(n^2)$, onde $n := r - p$.

Seleção em tempo linear

Como fazer algo melhor?

Seleção em tempo linear

Como fazer algo melhor?

Vamos usar de novo divisão e conquista.

Veremos o algoritmo BFPRT,
de Blum, Floyd, Pratt, Rivest e Tarjan.

Seleção em tempo linear

Como fazer algo melhor?

Vamos usar de novo divisão e conquista.

Veremos o algoritmo ~~BFPRT~~ MoM,
de Blum, Floyd, Pratt, Rivest e Tarjan,
também conhecido como "Median of Medians"

Seleção em tempo linear

Como fazer algo melhor?

Vamos usar de novo divisão e conquista.

Veremos o algoritmo ~~BFPRT~~ MoM,
de Blum, Floyd, Pratt, Rivest e Tarjan,
também conhecido como "Median of Medians"

Se o pivô do PARTICIONE for a mediana do vetor,
qual seria o consumo de tempo do SELECT?

Seleção em tempo linear

Como fazer algo melhor?

Vamos usar de novo divisão e conquista.

Veremos o algoritmo ~~BFPRT~~ MoM,
de Blum, Floyd, Pratt, Rivest e Tarjan,
também conhecido como "Median of Medians"

Se o pivô do PARTICIONE for A mediana do vetor,
qual seria o consumo de tempo do SELECT?

A Mediana

Para os propósitos desta aula, vamos redefinir mediana:

A mediana de $A[1\,..\,n]$ é o $\lfloor \frac{n+1}{2} \rfloor$-ésimo menor elemento de $A[1\,..\,n]$

A Mediana

Para os propósitos desta aula, vamos redefinir mediana:

A mediana de $A[1\,..\,n]$ é o $\lfloor \frac{n+1}{2} \rfloor$-ésimo menor elemento de $A[1\,..\,n]$

```
MEDIANA-ORD (A, p, r)      ▷ mediana de A[p .. r)
1  n ← r − p
2  devolva SELECT-ORD (A, p, r, ⌊(n+1)/2⌋)
```

A Mediana

Para os propósitos desta aula, vamos redefinir mediana:

A mediana de $A[1\,..\,n]$ é o $\lfloor \frac{n+1}{2} \rfloor$-ésimo menor elemento de $A[1\,..\,n]$

```
MEDIANA-ORD (A, p, r)      ▷ mediana de A[p .. r)
1  n ← r − p
2  devolva SELECT-ORD (A, p, r, ⌊(n+1)/2⌋)
```

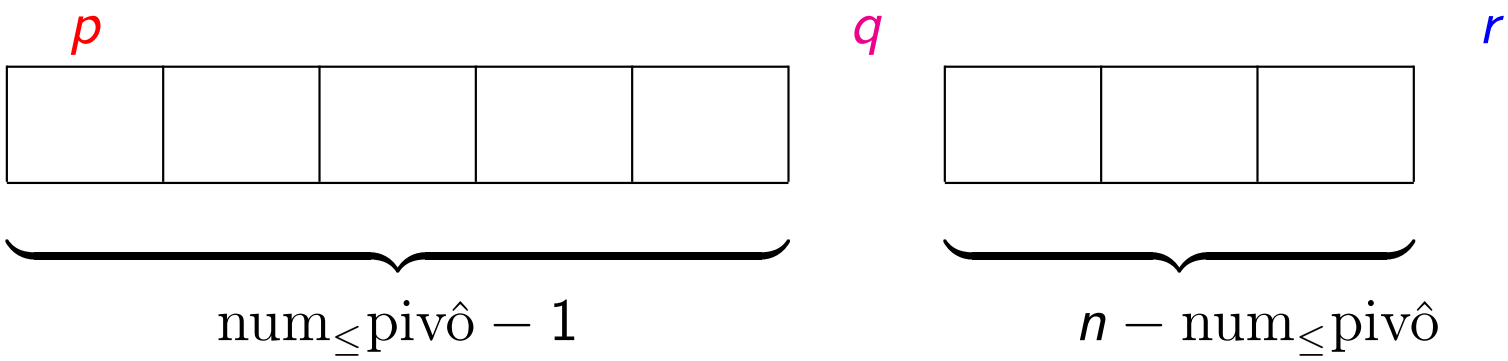
MEDIANA-ORD pode ser implementado de modo a consumir tempo $O(n \lg n)$, onde $n := r - p$.

Select-MoM

Recebe $A[p\,..\,r)$ e k tal que $1 \le k \le r-p$ e devolve o k-ésimo menor elemento de $A[p\,..\,r)$.

```
SELECT-MoM(A, p, r, k)
1  se r − p = 1
2      então devolva A[p]
3  q ← PARTICIONE-MoM (A, p, r)
4  num≤pivô ← q − p + 1      ▷ nº de elementos em A[p .. q]
5  se k = num≤pivô
6      então devolva A[q]
7  se k < num≤pivô
8      então devolva SELECT-MoM (A, p, q, k)
9      senão devolva SELECT-MoM (A, q+1, r, k−num≤pivô)
```

Particione-MoM



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

$$\max\{\text{num}_{\leq}\text{pivô} - 1, n - \text{num}_{\leq}\text{pivô}\} \leq \frac{7n}{10} + 3,$$

onde $n = r - p$ e $\text{num}_{\leq}\text{pivô} = q - p + 1$.

Suponha que $P(n) :=$ consumo de tempo máximo do algoritmo PARTICIONE-MoM quando $n = r - p$

Consumo de tempo

$T(n) :=$ consumo de tempo máximo do algoritmo SELECT-MoM quando $n = r - p$ e $i = \mathrm{num}_{\leq}\mathrm{pivô}$

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

$$\begin{aligned} T(n) &\leq \Theta(1) + P(n) + \max_i \{T(i-1), T(n-i)\} \\ &\leq \Theta(1) + P(n) + T(\lfloor \tfrac{7n}{10} \rfloor + 3) \end{aligned}$$

Lembre-se: $i - 1,\ n - i \leq \dfrac{7n}{10} + 3.$

Particione-MoM

$n := r - p$

Suponha que n é um múltiplo de 5.

PARTICIONE-MoM (A, p, r)

1 $n \leftarrow r - p$ ▷ Usamos vetor auxiliar $B[0 \,..\, \lceil n/5 \rceil)$
2 $n_B \leftarrow \lceil n/5 \rceil$
3 **para** j **em** $[0 \,..\, n_B)$ **faça**
4 $B[j] \leftarrow$ MEDIANA-ORD $(A, p + 5j, p + 5j + 5)$

Particione-MoM

$n := r - p$

Suponha que n é um múltiplo de 5.

PARTICIONE-MoM (A, p, r)

1 $n \leftarrow r - p$ ▷ Usamos vetor auxiliar $B[0\,..\,\lceil n/5 \rceil)$
2 $n_B \leftarrow \lceil n/5 \rceil$
3 **para** j **em** $[0\,..\,n_B)$ **faça**
4 $B[j] \leftarrow$ MEDIANA-ORD $(A, p + 5j, p + 5j + 5)$



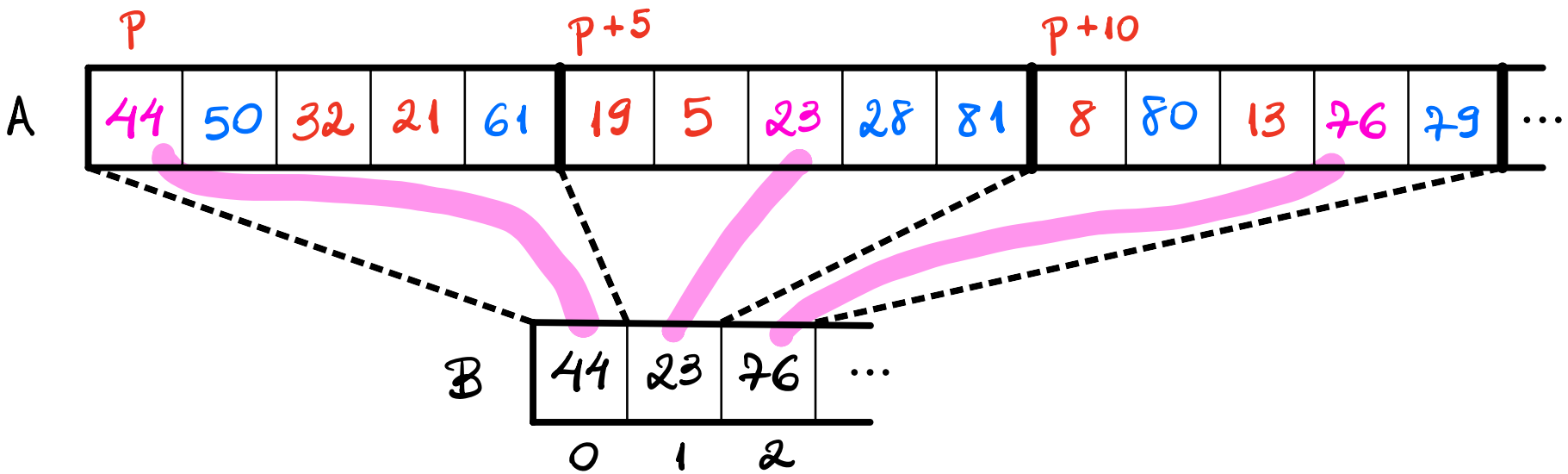
Particione-MoM

$n := r - p$

Suponha que n é um múltiplo de 5.

PARTICIONE-MoM (A, p, r)

1 $n \leftarrow r - p$ ▷ Usamos vetor auxiliar $B[0\,..\,\lceil n/5 \rceil)$
2 $n_B \leftarrow \lceil n/5 \rceil$
3 **para** j **em** $[0\,..\,n_B)$ **faça**
4 $B[j] \leftarrow$ MEDIANA-ORD $(A, p + 5j, p + 5j + 5)$



Particione-MoM

$n := r - p$

Suponha que n é um múltiplo de 5. Se não, troque $p + 5j + 5$ na linha 4 por $\min\{p + 5j + 5, r\}$.

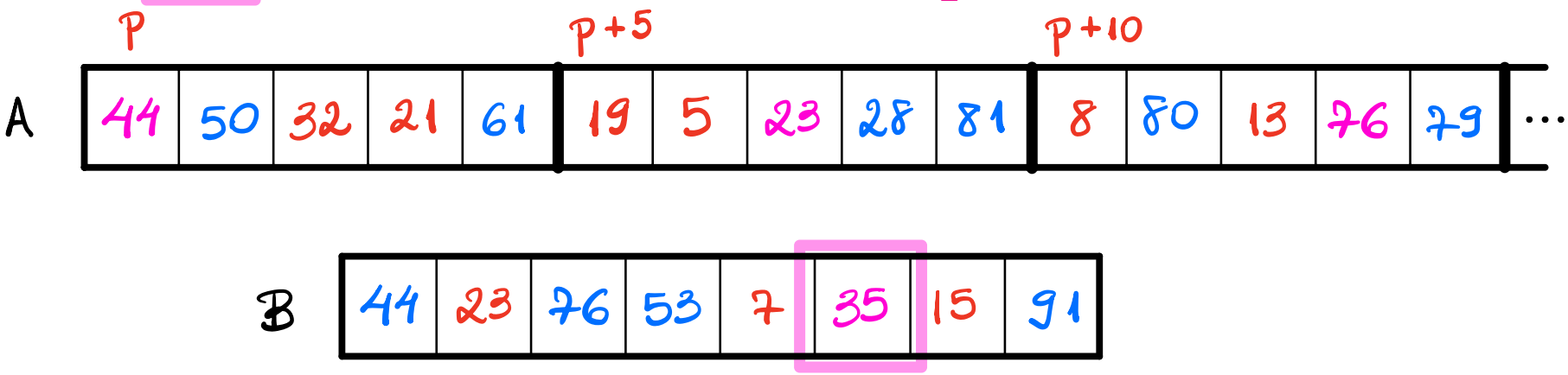
PARTICIONE-MoM (A, p, r)
1 $n \leftarrow r - p$ ▷ Usamos vetor auxiliar $B[0 \,..\, \lceil n/5 \rceil)$
2 $n_B \leftarrow \lceil n/5 \rceil$
3 **para** j **em** $[0 \,..\, n_B)$ **faça**
4 $B[j] \leftarrow$ MEDIANA-ORD $(A, p + 5j, p + 5j + 5)$

Particione-MoM

$n := r - p$

Suponha que n é um múltiplo de 5. Se não, troque $p + 5j + 5$ na linha 4 por $\min\{p + 5j + 5, r\}$.

PARTICIONE-MoM (A, p, r)
1 $n \leftarrow r - p$ ▷ Usamos vetor auxiliar $B[0 \,..\, \lceil n/5 \rceil)$
2 $n_B \leftarrow \lceil n/5 \rceil$
3 **para** j **em** $[0 \,..\, n_B)$ **faça**
4 $B[j] \leftarrow$ MEDIANA-ORD $(A, p + 5j, p + 5j + 5)$
5 MoM $\leftarrow$ SELECT-MoM $(B, 0, n_B, \lfloor \frac{n_B+1}{2} \rfloor)$ ▷ median of medians



Particione-MoM

$n := r - p$

Suponha que n é um múltiplo de 5. Se não, troque $p + 5j + 5$ na linha 4 por $\min\{p + 5j + 5, r\}$.

PARTICIONE-MoM (A, p, r)

1 $n \leftarrow r - p$ ▷ Usamos vetor auxiliar $B[0 \,..\, \lceil n/5 \rceil)$
2 $n_B \leftarrow \lceil n/5 \rceil$
3 **para** j **em** $[0 \,..\, n_B)$ **faça**
4 $B[j] \leftarrow$ MEDIANA-ORD $(A, p + 5j, p + 5j + 5)$
5 $\mathrm{MoM} \leftarrow$ SELECT-MoM $(B, 0, n_B, \lfloor \frac{n_B+1}{2} \rfloor)$ ▷ median of medians
6 Encontre ℓ em $[p, r)$ tal que $A[\ell] = \mathrm{MoM}$
7 $A[\ell] \leftrightarrow A[r - 1]$
8 **devolva** PARTICIONE (A, p, r)

Consumo de tempo do Particione-MoM

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

linha	consumo de todas as execuções da linha
1–2	$= \Theta(1)$
3–4	$= \lceil n/5 \rceil \, \Theta(1) \;=\; \Theta(n)$
5	$\leq T(\lceil n/5 \rceil)$
6	$= \Theta(n)$
7	$= \Theta(1)$
8	$= \Theta(n)$

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

Consumo de tempo do Select-MoM

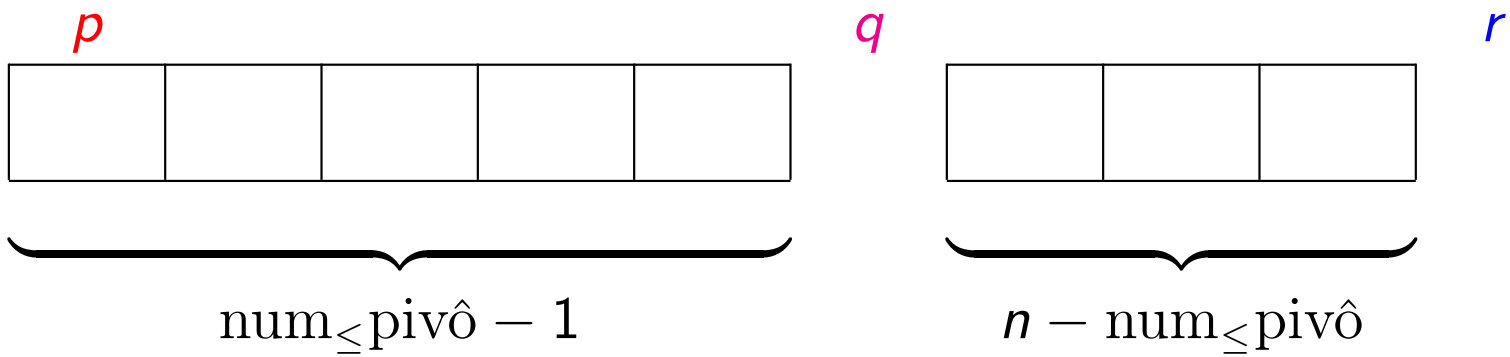
$T(n) :=$ consumo de tempo máximo do algoritmo SELECT-MoM quando $n = r - p$

Temos que

$$
\begin{aligned}
T(1) &= \Theta(1) \\
T(n) &\leq \Theta(1) + P(n) + T\left(\left\lfloor \frac{7n}{10} \right\rfloor + 3\right) \\
&\leq \Theta(1) + \Theta(n) + T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\left\lfloor \frac{7n}{10} \right\rfloor + 3\right) \\
&= \Theta(n) + T\left(\left\lceil \frac{2n}{10} \right\rceil\right) + T\left(\left\lfloor \frac{7n}{10} \right\rfloor + 3\right)
\end{aligned}
$$

para todo $n \geq 2$.

Particione-MoM

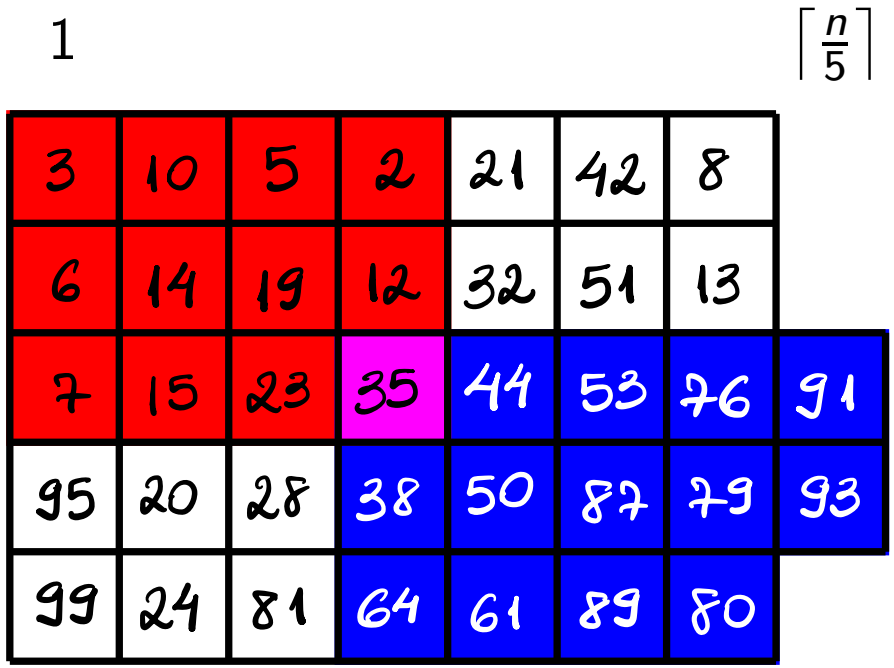


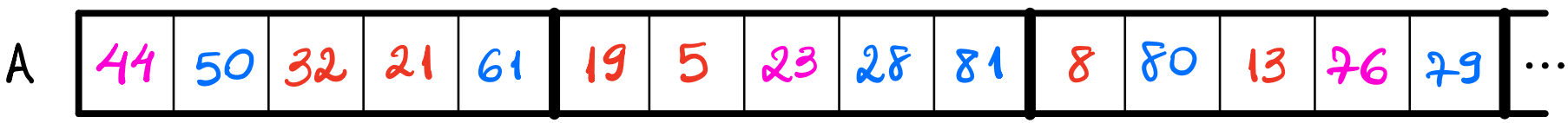
Rearranja $A[p\,..\,r)$ e devolve um índice q, com $q \in [p\,..\,r)$, tal que $A[p\,..\,q) \leq A[q] < A[q{+}1\,..\,r)$ e

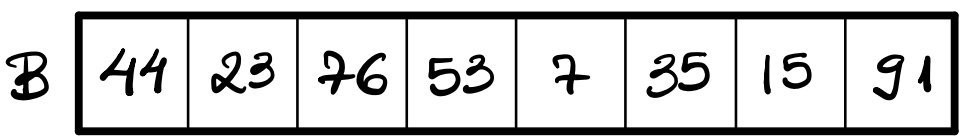
$$\max\{\mathrm{num}_{\leq}\mathrm{pivô} - 1, n - \mathrm{num}_{\leq}\mathrm{pivô}\} \leq \frac{7n}{10} + 3,$$

onde $n = r - p$ e $\mathrm{num}_{\leq}\mathrm{pivô} = q - p + 1$.

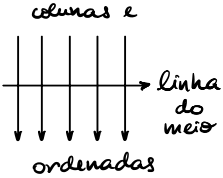
Particione-MoM







Particione-MoM

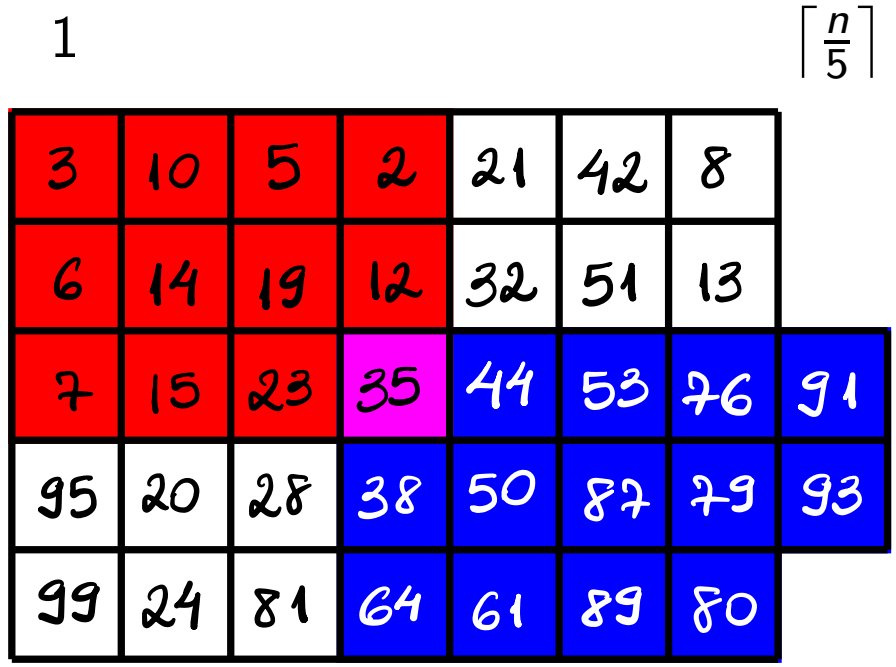


1							$\left\lceil \frac{n}{5} \right\rceil$
3	10	5	2	21	42	8	
6	14	19	12	32	51	13	
7	15	23	35	44	53	76	91
95	20	28	38	50	87	79	93
99	24	81	64	61	89	80	

A															
44	50	32	21	61	19	5	23	28	81	8	80	13	76	79	...

B							
44	23	76	53	7	35	15	91

Particione-MoM



7 **se** $k < \text{num}_{\leq}\text{pivô}$

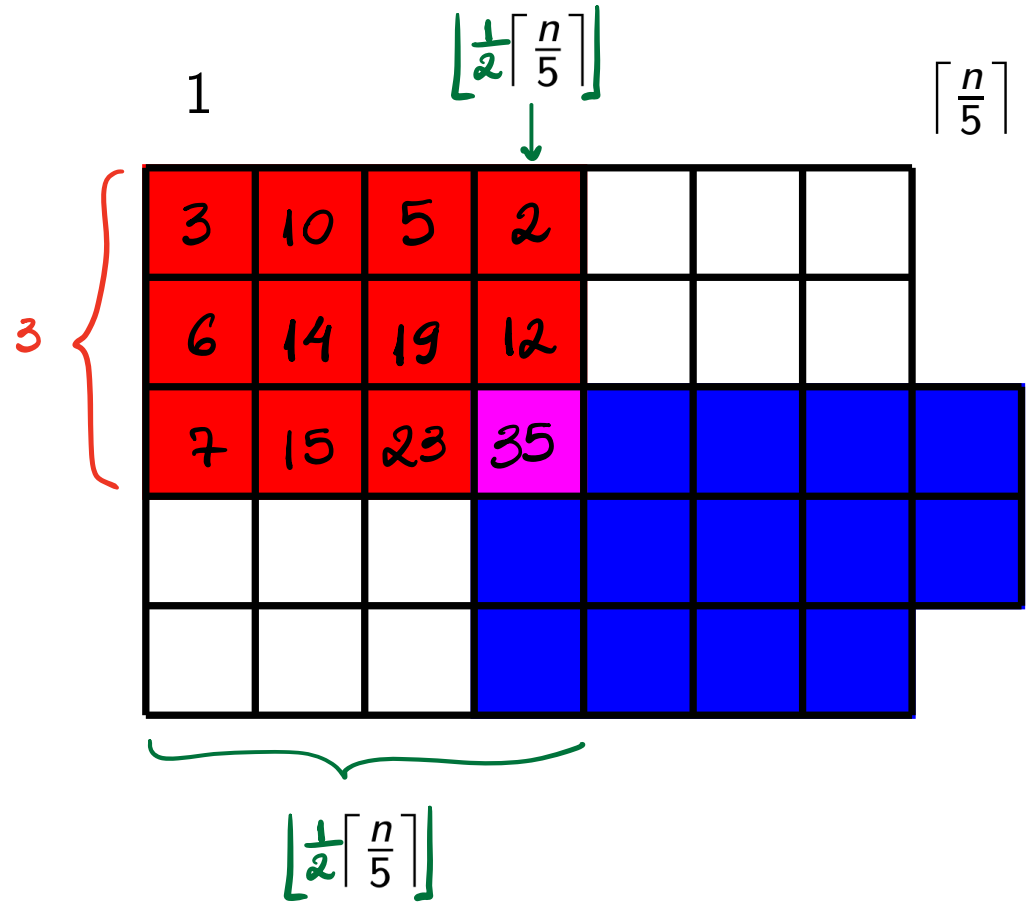
8 **então devolva** SELECT-MoM (A, p, q, k)

9 **senão devolva** SELECT-MoM $(A, q+1, r, k-\text{num}_{\leq}\text{pivô})$

"deleta" região azul (incl. pivô)

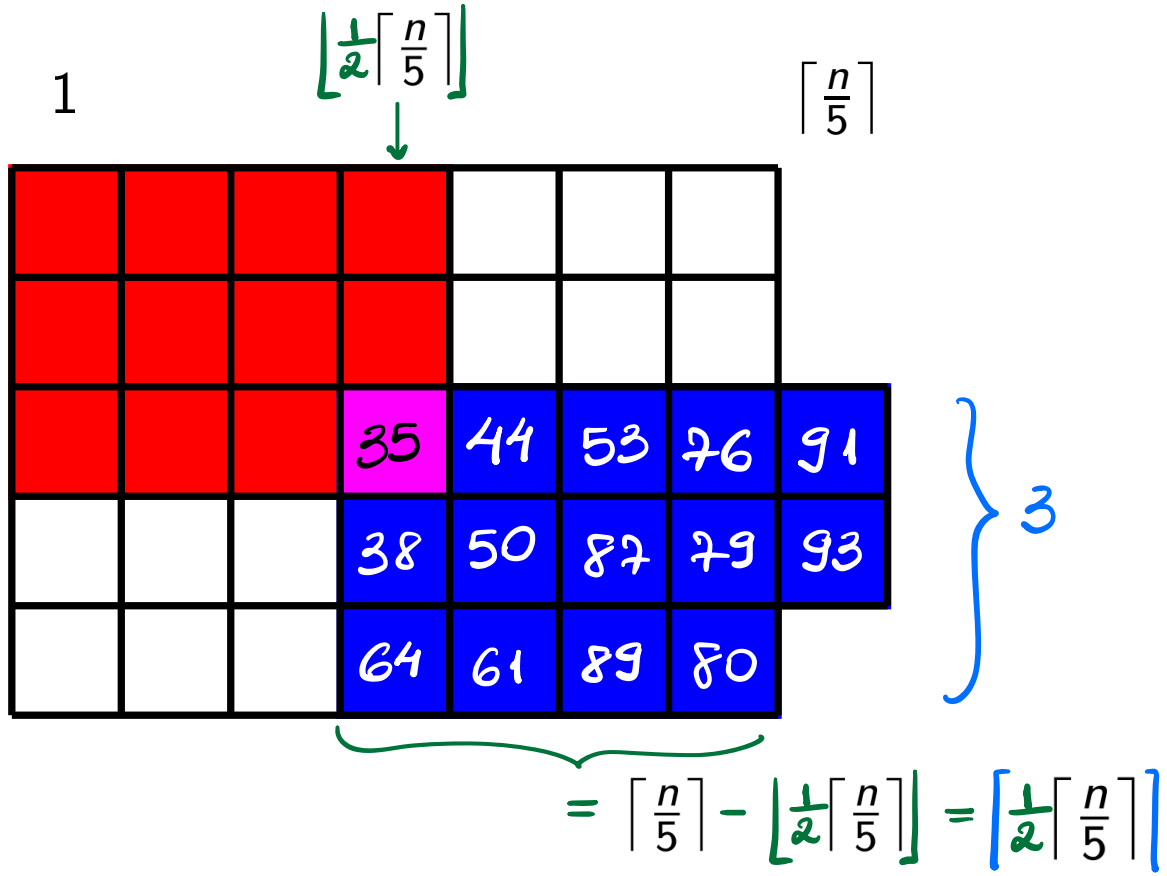
"deleta" região vermelha (incl. pivô)

Particione-MoM



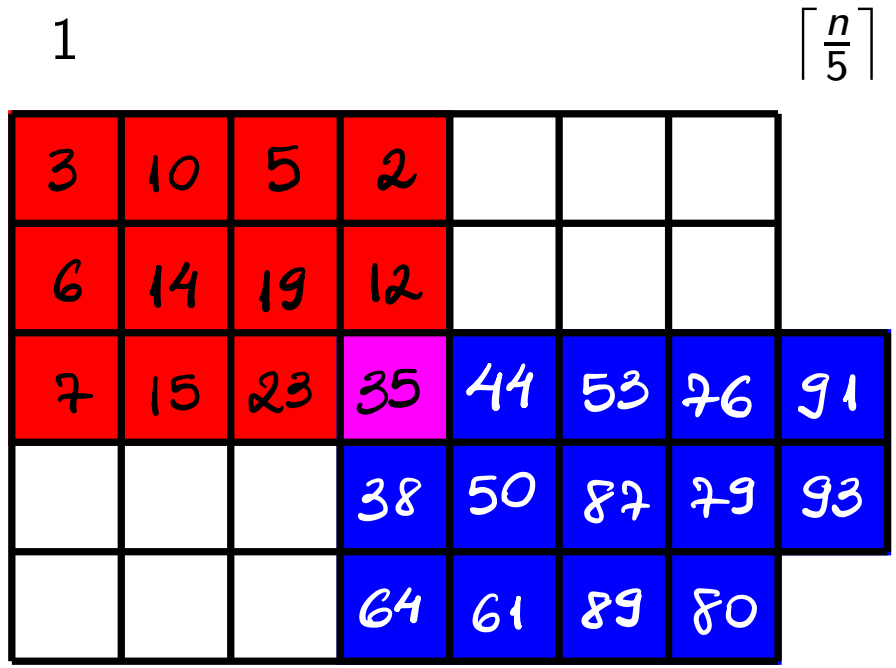
$$\text{vermelhos} = 3 \cdot \left\lfloor \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rfloor \geq 3 \cdot \left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 1 \right) = 3\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 3$$

Particione-MoM



azuis $\geq 3 \cdot \lceil \frac{1}{2} \lceil \frac{n}{5} \rceil \rceil \geq 3 \cdot \lceil \frac{1}{2} \lceil \frac{n}{5} \rceil \rceil - 3$

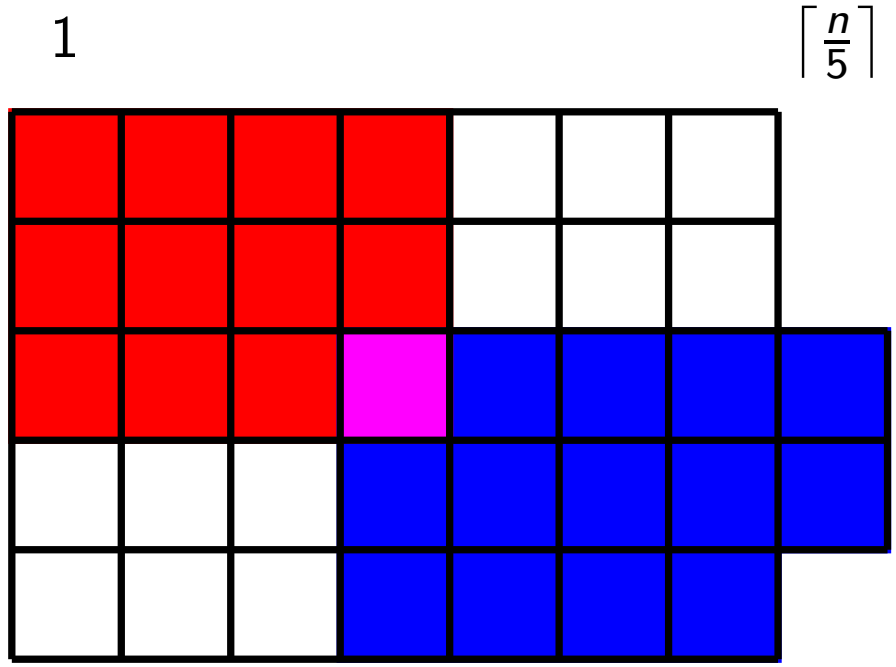
Particione-MoM



$$\text{vermelhos} \geq 3\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 3$$

$$\text{azuis} \geq 3 \cdot \left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 3$$

Particione-MoM

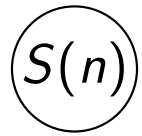


$$\max\{\text{num}_{\leq}\text{pivô} - 1, n - \text{num}_{\leq}\text{pivô}\} \leq n - \left(3\left\lceil\frac{1}{2}\left\lceil\frac{n}{5}\right\rceil\right\rceil - 3\right)$$

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

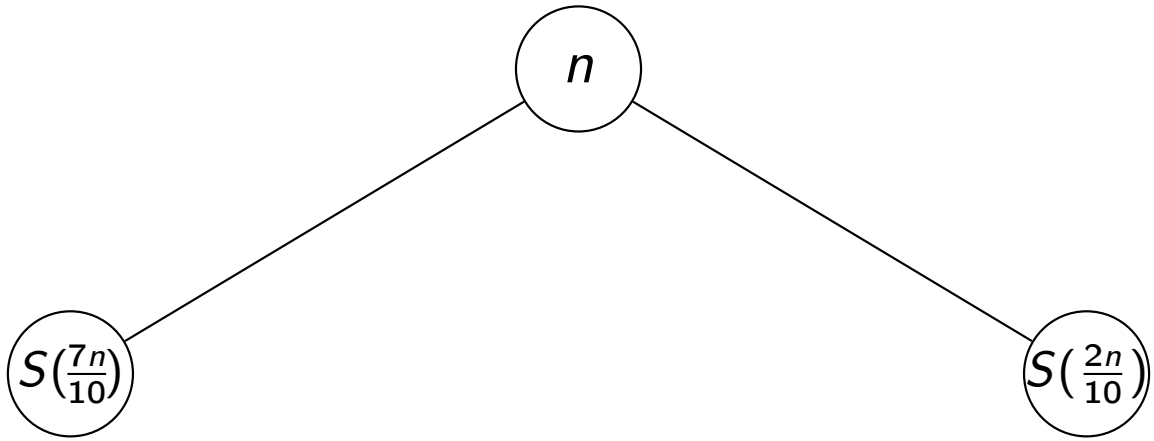
Um chute para a classe O?

Árvore da recorrência:



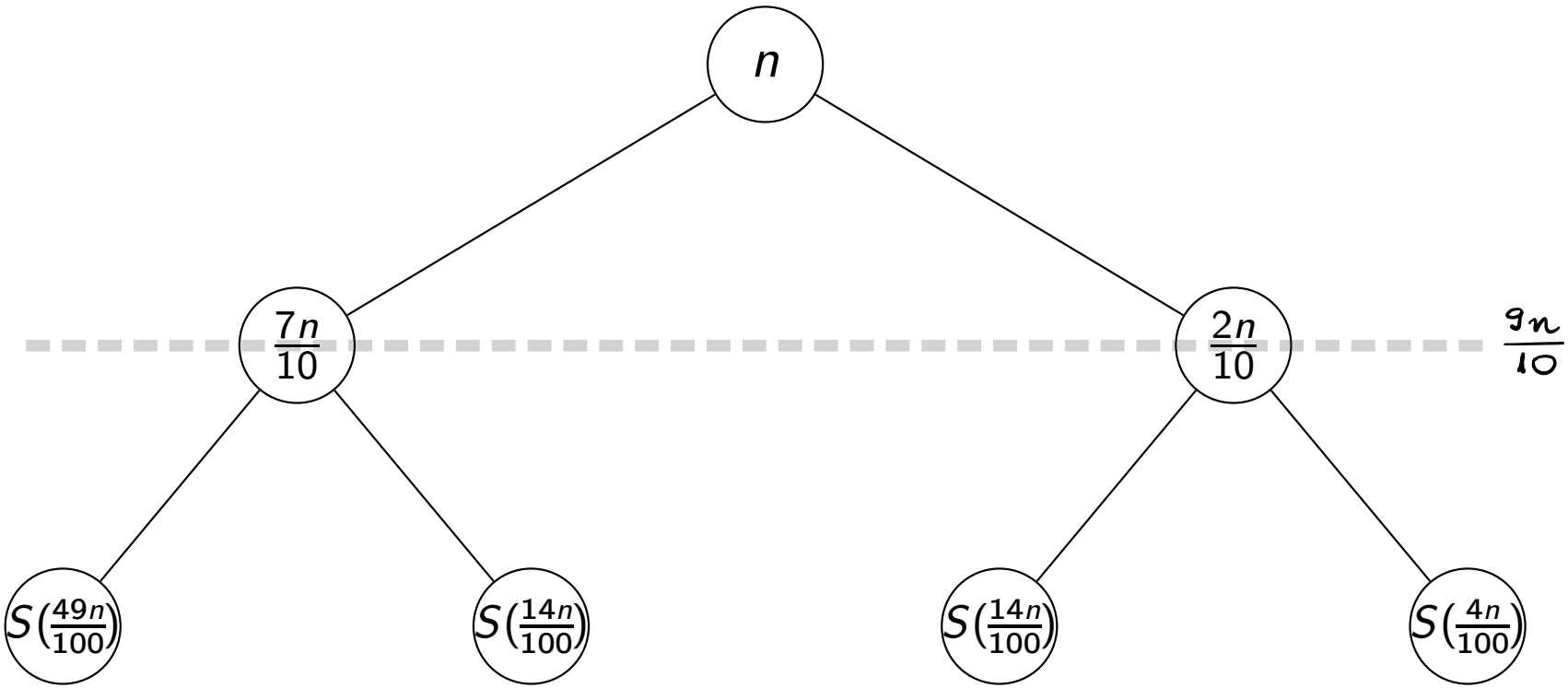
Um chute para a classe O?

Árvore da recorrência:



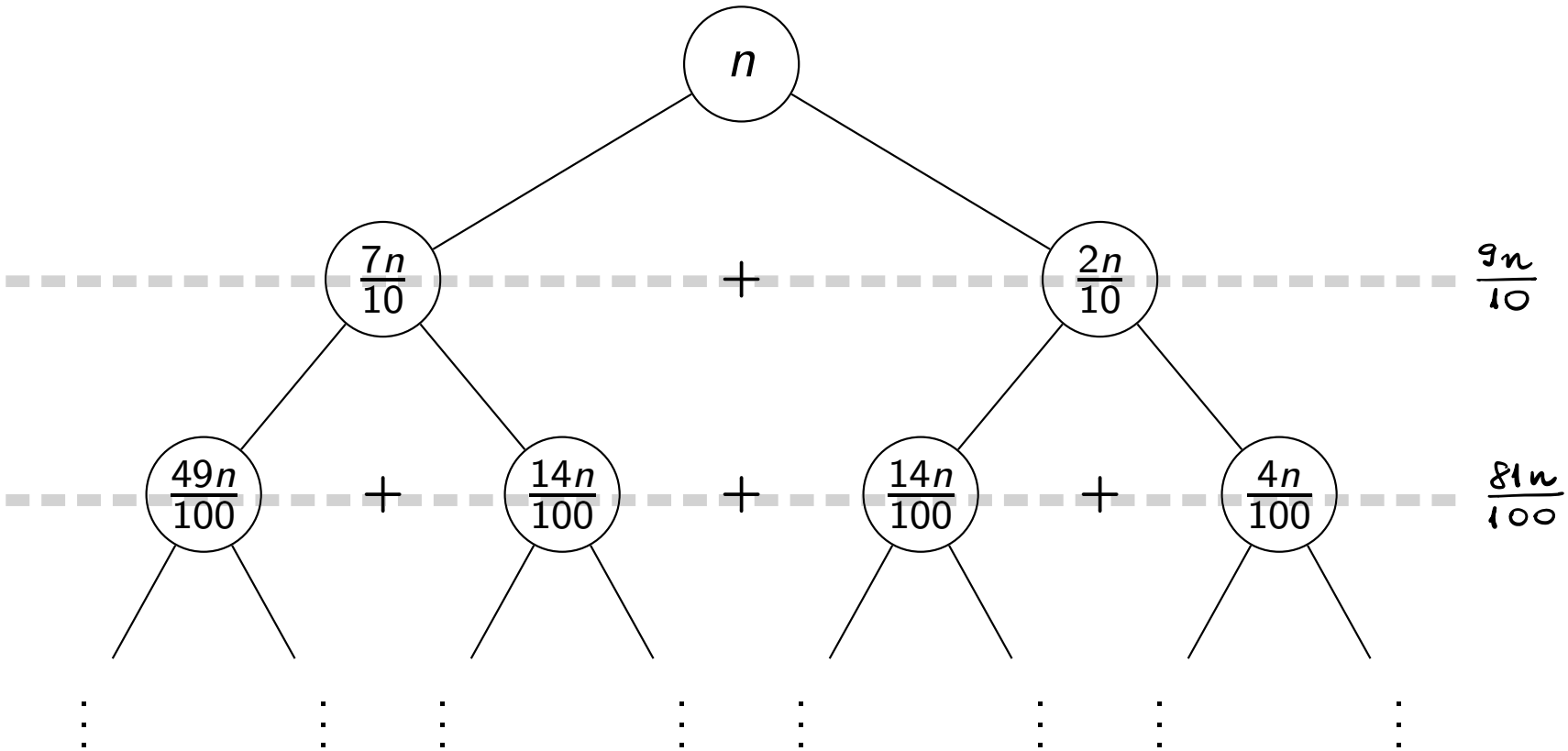
Um chute para a classe O?

Árvore da recorrência:



Um chute para a classe O?

Árvore da recorrência:



Contas

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

Contas

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

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

Contas

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

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

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

Consumo de tempo do Select-MoM

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

$$S(n) = 1 \quad \text{para todo } n < 100$$

$$S(n) = S\left(\left\lceil \frac{n}{5} \right\rceil\right) + S\left(\left\lfloor \frac{7n}{10} \right\rfloor + 3\right) + n \quad \text{para todo } n \geq 100$$

Vamos verificar que $S(n) \leq C\,n$ para todo $n \geq 1$, onde $C := 20$.

Prova: Se $n \in \{1, \ldots, 99\}$, então $S(n) = 1 < 20 \leq C\,n$.

Recorrência

Se $n \geq 100$, então

$$
\begin{aligned}
S(n) &\leq S\left(\left\lceil \frac{n}{5} \right\rceil\right) + S\left(\left\lfloor \frac{7n}{10} \right\rfloor + 3\right) + n \\
&\overset{\text{hi}}{\leq} C \cdot \left\lceil \frac{n}{5} \right\rceil + C \cdot \left(\left\lfloor \frac{7n}{10} \right\rfloor + 3\right) + n \\
&\leq C\left(\frac{n}{5} + 1\right) + C\left(\frac{7n}{10} + 3\right) + n \\
&= \frac{C}{5}n + C + \frac{7C}{10}n + 3C + n \\
&= \frac{2C}{10}n + \frac{7C}{10}n + n + 4C \\
&= \left(\frac{9C}{10} + 1\right)n + 4C \\
&\leq Cn \quad (\text{pois } n \geq 100 \text{ e } C = 20).
\end{aligned}
$$

Logo, $T(n)$ é $\mathrm{O}(n)$.

Conclusão

O consumo de tempo do SELECT-MoM é $O(n)$.