Min-Max com $\lfloor 3n/2 \rfloor$ comparações (ao invés de $2n - 2$)

MIN-MAX(A, n) para n par

menor $\leftarrow \min\{A[1], A[2]\}$; *maior* $\leftarrow \max\{A[1], A[2]\}$
para $i \leftarrow 2$ **até** $n/2$ **faça**:
    $x \leftarrow \min\{A[2i-1], A[2i]\}$; $X \leftarrow \max\{A[2i-1], A[2i]\}$
    menor $\leftarrow \min\{$*menor*$, x\}$; *maior* $\leftarrow \max\{$*maior*$, X\}$
retorne (*menor*, *maior*)

MIN-MAX(A, n) para n ímpar

menor $\leftarrow A[1]$; *maior* $\leftarrow A[1]$
para $i \leftarrow 1$ **até** $(n-1)/2$ **faça**:
    $x \leftarrow \min\{A[2i], A[2i+1]\}$; $X \leftarrow \max\{A[2i], A[2i]\}$
    menor $\leftarrow \min\{$*menor*$, x\}$; *maior* $\leftarrow \max\{$*maior*$, X\}$
retorne (*menor*, *maior*)

SELEÇÃO-Simples(A, n, k)

$k \leftarrow \min\{k, n\}$; InsertionSort$(A, 1, k)$
para $j \leftarrow k+1$ **até** n **faça**:
    $chave \leftarrow A[j]$; $A[j] \leftarrow A[k+1]$
    $i \leftarrow k$
    enquanto $i \geq 1$ **e** $A[i] > chave$ **faça**
        $A[i+1] \leftarrow A[i]$; $i \leftarrow i-1$
    $A[i+1] \leftarrow chave$
retorne $A[k]$

Tempo do SELEÇÃO-Simples: $O(k \cdot n)$

- **Se** $k = O(1) \implies$ **tempo** $O(n)$
- **Se** $k = O(\log \log n) \implies$ **tempo** $O(n \log \log n)$
- **Se** $k = O(\log n) \implies$ **tempo** $O(n \log n)$
- **Se** $k = O(n) \implies$ **tempo** $O(n^2)$

SELECT-aleat(A, p, r, k)

```
se (n ≤ 20) então retorne força-bruta(A, p, r, k)
q ← PARTICIONE-aleat(A, p, r)
se k = q − p + 1 então
        retorne A[q]
senão se k < q − p + 1 então
        retorne SELECT-aleat (A, p, q − 1, k)
senão retorne SELECT-aleat(A, q + 1, r, k − (q − p + 1))
```

onde $n \; := \; r - p + 1$ (tamanho do subvetor).

Tempo do QUICKSORT-ALE (pior caso)

$$T(n) = T(n-1) + T(0) + c \cdot n$$
$$T(n) = T(n-10) + T(9) + c \cdot n$$
$$T(n) = T(n-100) + T(99) + c \cdot n$$

- **Tempo:** $O(n^2)$ [árvore de recursão]

Tempo do SELECT-ALE (pior caso)

$$T(n) = T(n-1) + c \cdot n$$
$$T(n) = T(n-10) + c \cdot n$$
$$T(n) = T(n-100) + c \cdot n$$

- **Tempo:** $O(n^2)$ [árvore de recursão]

Tempo do QUICKSORT-ALE (intuição para o caso médio)

$$T(n) = T(\lfloor 9n/10 \rfloor) + T(\lfloor n/10 \rfloor) + c \cdot n$$
$$T(n) = T(\lfloor 99n/100 \rfloor) + T(\lfloor n/100 \rfloor) + c \cdot n$$
$$T(n) = T(\lfloor 999n/1000 \rfloor) + T(\lfloor n/1000 \rfloor) + c \cdot n$$

- **Tempo:** $O(n \log n)$ [árvore de recursão]
- Ou por indução: $T(n) \leq cn \log_{10/9} n$ para $T(1) = T(0) = 0$

Tempo do SELECT-ALE (intuição para o caso médio)

$$T(n) = T(\lfloor 9n/10 \rfloor) + c \cdot n$$
$$T(n) = T(\lfloor 99n/100 \rfloor) + c \cdot n$$
$$T(n) = T(\lfloor 999n/1000 \rfloor) + c \cdot n$$

- **Tempo:** $O(n)$ [método mestre/árvore de recursão]
- Ou por indução: $T(n) \leq 10 \cdot cn$ para $T(1) = T(0) = 0$

SELECT-aleat(A, p, r, k)

se $(n \leq 20)$ **então retorne** força-bruta(A, p, r, k)
$q \leftarrow$ PARTICIONE-aleat(A, p, r)
se $k = q - p + 1$ **então**
        retorne $A[q]$
senão se $k < q - p + 1$ **então**
        retorne SELECT-aleat $(A, p, q - 1, k)$
senão retorne SELECT-aleat$(A, q + 1, r, k - (q - p + 1))$

onde $n \;:=\; r - p + 1$ (tamanho do subvetor).

SELECT-linear(A, p, r, k)

1 **se** $(n \leq 20)$ **então retorne** força-bruta(A, p, r, k)

1.1 **Crie** vetor M com $\lceil n/5 \rceil$ elementos

1.2 **para** $i \leftarrow 0$ **até** $\lceil n/5 \rceil - 1$

1.3   **Insertion-Sort**$(A, p + 5i, p + 5i + 4)$;

1.4 $M[i + 1] \leftarrow A[p + 5i + 2]$

1.5 **pivo** $\leftarrow$ SELECT-linear$(M, 1, \lceil n/5 \rceil, \lceil n/10 \rceil)$;   **apaga** M

2 $q \leftarrow$ PARTICIONE$(A, p, r,$ **pivo**$)$

3 **se** $k = q - p + 1$ **então**

4   **retorne** $A[q]$

5 **senão se** $k < q - p + 1$ **então**

6   **retorne** SELECT-linear $(A, p, q - 1, k)$

7 **senão retorne** SELECT-linear$(A, q + 1, r, k - (q - p + 1))$

Grupos de 5 elementos: M tem $\lceil n/5 \rceil$ elementos

- $n/10$ M-elem. $\leq$ *pivo* $\implies$ $\geq 3 \cdot n/10$ A-elem. $\leq$ *pivo*
- $n/10$ M-elem. $\geq$ *pivo* $\implies$ $\geq 3 \cdot n/10$ A-elem. $\geq$ *pivo*
- Então PARTICIONE divide A em pelo menos $3n/10$ e $7n/10$

Grupos de 7 elementos: M tem $\lceil n/7 \rceil$ elementos

- $n/14$ M-elem. $\leq$ *pivo* $\implies$ $\geq 4 \cdot n/14$ A-elem. $\leq$ *pivo*
- $n/14$ M-elem. $\geq$ *pivo* $\implies$ $\geq 4 \cdot n/14$ A-elem. $\geq$ *pivo*
- Então PARTICIONE divide A em pelo menos $2n/7$ e $5n/7$

Grupos de 3 elementos: M tem $\lceil n/3 \rceil$ elementos

- $n/6$ M-elem. $\leq$ *pivo* $\implies$ $\geq 2 \cdot n/6$ A-elem. $\leq$ *pivo*
- $n/6$ M-elem. $\geq$ *pivo* $\implies$ $\geq 2 \cdot n/6$ A-elem. $\geq$ *pivo*
- Então PARTICIONE divide A em pelo menos $n/3$ e $2n/3$

Grupos de 5 elementos: M tem $\lceil n/5 \rceil$ elementos

- Então PARTICIONE divide A em pelo menos $3n/10$ e $7n/10$
- $T(n) \;=\; T(n/5) \;+\; T(7n/10) \;+\; \Theta(n)$
- Árvore de recursão: $T(n) \;=\; \Theta(n)$

Grupos de 7 elementos: M tem $\lceil n/7 \rceil$ elementos

- Então PARTICIONE divide A em pelo menos $2n/7$ e $5n/7$
- $T(n) \;=\; T(n/7) \;+\; T(5n/7) \;+\; \Theta(n)$
- Árvore de recursão: $T(n) \;=\; \Theta(n)$

Grupos de 3 elementos: M tem $\lceil n/3 \rceil$ elementos

- Então PARTICIONE divide A em pelo menos $n/3$ e $2n/3$
- $T(n) \;=\; T(n/3) \;+\; T(2n/3) \;+\; \Theta(n)$
- Árvore de recursão: $T(n) \;=\; \Theta(n \log n)$