# Aula 09: Algoritmos de ordenação em arranjos

## Ordenação por fusão

David Déharbe
Programa de Pós-graduação em Sistemas e Computação
Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemática Aplicada

Download me from `http://DavidDeharbe.github.io`.

# Plano da aula

- Fusão de faixas contíguas ordenadas
  - Definição
  - Algoritmo
  - Simulação
  - Complexidade
  - Correção
- Ordenação por fusão
  - Algoritmo
  - Simulação
  - Complexidade
  - Complexidade
  - Verificação

- Escreva um algoritmo $\textsc{Merge}(A, l, m, u)$ tal que
  - $A$ é um arranjo,
  - $l$, $m$, e $u$ são três posições do arranjo, tais que $l \leq m \leq u$,
  - as sub-faixas $l \mathinner{.\,.} m$ e $m+1 \mathinner{.\,.} u$ são ordenadas.
- O resultado do algoritmo é que
  - a sub-faixa $l \mathinner{.\,.} u$ de $A$ contenha os elementos inicialmente nas sub-faixas $l \mathinner{.\,.} m$ e $m+1 \mathinner{.\,.} u$;
  - a sub-faixa $l \mathinner{.\,.} u$ de $A$ esteja ordenada;
  - as sub-faixas $1 \mathinner{.\,.} l-1$ e $u+1 \mathinner{.\,.} n$ permanecem inalteradas.
- Restrição: o algoritmo deve ter complexidade linear.

# Fusão de faixas ordenadas

## Um algoritmo

```
MERGE(A, l, m, u)
i = 1, j = l, k = m + 1
while j ≤ m and k ≤ u
    if A[j] ≤ A[k]
        B[i] = A[j], j = j + 1, i = i + 1
    else B[i] = A[k], k = k + 1, i = i + 1
while j ≤ m
    B[i] = A[j], j = j + 1, i = i + 1
while k ≤ u
    B[i] = A[k], k = k + 1, i = i + 1
for i = l to u
    A[i] = B[i − l + 1]
```

# Fusão de faixas ordenadas

## Simulação

```
MERGE(A, l, m, u)
i = 1, j = l, k = m + 1
while j ≤ m and k ≤ u
    if A[j] ≤ A[k]
        B[i] = A[j], j = j + 1, i = i + 1
    else B[i] = A[k], k = k + 1, i = i + 1
while j ≤ m
    B[i] = A[j], j = j + 1, i = i + 1
while k ≤ u
    B[i] = A[k], k = k + 1, i = i + 1
for i = l to u
    A[i] = B[i − l + 1]
```

$$A = \langle 2, 4, 5, 5, 1, 2, 3, 5 \rangle, l = 1, m = 4, u = 8$$

- Cada um dos três primeiros laços ou incrementa $j$, ou incrementa $k$.
- Há, exatamente, $m - l$ incrementos de $j$ e $u - (m + 1)$ incrementos de $k$
- O custo acumulado dos três primeiros laços é $\Theta(m - l + u - (m + 1)) = \Theta(u - l)$
- O custo do quarto laço é $\Theta(u - l)$
- Complexidade: $\Theta(u - l)$.

### Definição (*sorted*)

$$sorted(A, i, j) \equiv \forall k | i \leq k < j \cdot A[k] \leq A[k+1]$$

## Definição (*sorted*)

$$sorted(A, i, j) \equiv \forall k | i \leq k < j \cdot A[k] \leq A[k+1]$$

$\textsc{Merge}(A, l, m, u)$

- pré-condição: $A = \langle a_1, \ldots, a_n \rangle \wedge 1 \leq l, m, u \leq n \wedge$
  $sorted(A, l, m) \wedge sorted(A, m+1, u)$
- pós-condição
  $A[1..l-1] = \langle a_1, \ldots, a_{l-1} \rangle \wedge A[u+1..n] = \langle a_{u+1}, \ldots, a_n \rangle$
  $A[l..u] \in permutation(\langle a_l, \ldots, a_u \rangle) \wedge sorted(A, l, u)$

# Ordenação por fusão *merge sort*

## Algoritmo

(Cormen et al. 1990)

```
MERGE-SORT(A, l, u)
if l < u
    m = l + (u − l)/2
    MERGE-SORT(A, l, m)
    MERGE-SORT(A, m + 1, u)
    MERGE(A, l, m, u)
```

$\Longrightarrow$ $\textsc{Merge-Sort}(A, 1, \mathit{length}(A))$

$\textsc{Merge-Sort}(A, l, u)$

```
if l < u
    m = l + (u − l)/2
    MERGE-SORT(A, l, m)
    MERGE-SORT(A, m + 1, u)
    MERGE(A, l, m, u)
```

$\Longrightarrow \textsc{Merge-Sort}(A, 1, \mathit{length}(A))$

$$A = \langle 5, 2, 4, 6, 1, 3, 2, 6 \rangle$$

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

- $T(n) = 2 \times T(n/2) + \Theta(n)$
- Teorema master

- $T(n) = 2 \times T(n/2) + \Theta(n)$
- Teorema master
- $T(n) = aT(\frac{n}{b}) + f(n)$, onde $a \geq 1, b \geq 1$;
- Se existe $k \geq 0$ tal que $f(n) \in \Theta(n^c \log^k n)$ onde $c = \log_b a$ então $T(n) \in \Theta(n^c \log^{k+1} n)$.

- $T(n) = 2 \times T(n/2) + \Theta(n)$
- Teorema master
- $T(n) = aT(\frac{n}{b}) + f(n)$, onde $a \geq 1, b \geq 1$;
- Se existe $k \geq 0$ tal que $f(n) \in \Theta(n^c \log^k n)$ onde $c = \log_b a$ então $T(n) \in \Theta(n^c \log^{k+1} n)$.
- $c = \log_2 2 = 1$

# Ordenação por fusão

## Complexidade

- $T(n) = 2 \times T(n/2) + \Theta(n)$
- Teorema master
- $T(n) = aT(\frac{n}{b}) + f(n)$, onde $a \geq 1, b \geq 1$;
- Se existe $k \geq 0$ tal que $f(n) \in \Theta(n^c \log^k n)$ onde $c = \log_b a$ então $T(n) \in \Theta(n^c \log^{k+1} n)$.
- $c = \log_2 2 = 1$
- $n = n^1 \times \log^0 n$, logo $k = 0$ satisfaz a condição.

- $T(n) = 2 \times T(n/2) + \Theta(n)$
- Teorema master
- $T(n) = aT(\frac{n}{b}) + f(n)$, onde $a \geq 1, b \geq 1$;
- Se existe $k \geq 0$ tal que $f(n) \in \Theta(n^c \log^k n)$ onde $c = \log_b a$ então $T(n) \in \Theta(n^c \log^{k+1} n)$.
- $c = \log_2 2 = 1$
- $n = n^1 \times \log^0 n$, logo $k = 0$ satisfaz a condição.
- Então $T(n) \in \Theta(n \log n)$.

```
MERGE-SORT(A, l, u)
if l < u
        m = l + (u − l)/2
        MERGE-SORT(A, l, m)
        MERGE-SORT(A, m + 1, u)
        MERGE(A, l, m, u)
```

- pré-condição
- pós-condição

# Ordenação por fusão

## Correção

```
MERGE-SORT(A, l, u)
if l < u
    m = l + (u − l)/2
    MERGE-SORT(A, l, m)
    MERGE-SORT(A, m + 1, u)
    MERGE(A, l, m, u)
```

- pré-condição $A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n$
- pós-condição

## Ordenação por fusão

### Correção

```
MERGE-SORT(A, l, u)
if l < u
    m = l + (u − l)/2
    MERGE-SORT(A, l, m)
    MERGE-SORT(A, m + 1, u)
    MERGE(A, l, m, u)
```

- pré-condição $A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n$
- pós-condição
  $A[1 \mathinner{.\,.} l-1] = \langle a_1, \ldots a_{l-1} \rangle \wedge A[u+1 \mathinner{.\,.} n] = \langle a_{u+1}, \ldots a_n \rangle \wedge$
  $A[l \mathinner{.\,.} u] \in \mathit{permutation}(\langle a_l, \ldots, a_u \rangle) \wedge \mathit{sorted}(A, l, u)$

# Ordenação por fusão *merge sort*

## Verificação

```
MERGE-SORT(A, l, u)
if l < u
    m = l + (u − l)/2
    MERGE-SORT(A, l, m)
    MERGE-SORT(A, m + 1, u)
    MERGE(A, l, m, u)
```

# Ordenação por fusão *merge sort*

## Verificação

$\Gamma_1$ : $Pre(\text{MERGE-SORT}(A, l, u)) \wedge l \geq u \Rightarrow$
$Pos(\text{MERGE-SORT}(A, l, u))$

```
MERGE-SORT(A, l, u)
if l < u
    m = l + (u − l)/2
    MERGE-SORT(A, l, m)
    MERGE-SORT(A, m + 1, u)
    MERGE(A, l, m, u)
```

```
MERGE-SORT(A, l, u)
if l < u
    m = l + (u − l)/2
    A_0
    MERGE-SORT(A, l, m)
    A_1
    MERGE-SORT(A, m + 1, u)
    A_2
    MERGE(A, l, m, u)
    A_3
```

$\textsc{Merge-Sort}(A, l, u)$

**if** $l < u$

$\quad m = l + (u - l)/2$

$\quad A_0$

$$\Gamma_2 : \quad l < u \wedge m = l + \tfrac{u-l}{2} \wedge Pre(\textsc{Merge-Sort}(A_0, l, u)) \Rightarrow Pre(\textsc{Merge-Sort}(A_0, l, m))$$

$\quad \textsc{Merge-Sort}(A, l, m)$

$\quad A_1$

$\quad \textsc{Merge-Sort}(A, m + 1, u)$

$\quad A_2$

$\quad \textsc{Merge}(A, l, m, u)$

$\quad A_3$

# Ordenação por fusão *merge sort*

## Verificação

$\textsc{Merge-Sort}(A, l, u)$

**if** $l < u$

$\quad m = l + (u - l)/2$

$\quad A_0$

$\quad \textsc{Merge-Sort}(A, l, m)$

$\quad A_1$

$$
\begin{aligned}
\Gamma_3 : \quad & l < u \wedge m = l + \tfrac{u-l}{2} \wedge \\
& Pre(\textsc{Merge-Sort}(A_0, l, u)) \wedge \\
& Pos(\textsc{Merge-Sort}(A_1, l, m)) \\
\Rightarrow \quad & Pre(\textsc{Merge-Sort}(A_1, m+1, u))
\end{aligned}
$$

$\quad \textsc{Merge-Sort}(A, m+1, u)$

$\quad A_2$

$\quad \textsc{Merge}(A, l, m, u)$

$\quad A_3$

# Ordenação por fusão *merge sort*

## Verificação

$\textsc{Merge-Sort}(A, l, u)$

```
if l < u
    m = l + (u − l)/2
    A_0
    MERGE-SORT(A, l, m)
    A_1
    MERGE-SORT(A, m + 1, u)
    A_2
```

$$
\Gamma_4 : \quad \begin{array}{l}
l < u \wedge m = l + \frac{u-l}{2} \wedge \\
Pre(\textsc{Merge-Sort}(A_0, l, u)) \wedge \\
Pos(\textsc{Merge-Sort}(A_1, l, m)) \wedge \\
Pos(\textsc{Merge-Sort}(A_2, m+1, u)) \\
\Rightarrow \; Pre(\textsc{Merge}(A_2, l, m, u))
\end{array}
$$

```
    MERGE(A, l, m, u)
    A_3
```

# Ordenação por fusão *merge sort*

## Verificação

$\textsc{Merge-Sort}(A, l, u)$

**if** $l < u$

$\quad m = l + (u - l)/2$

$\quad A_0$

$\quad \textsc{Merge-Sort}(A, l, m)$

$\quad A_1$

$\quad \textsc{Merge-Sort}(A, m + 1, u)$

$\quad A_2$

$\quad \textsc{Merge}(A, l, m, u)$

$\quad A_3$

$$
\begin{array}{rl}
\Gamma_5 : & l < u \wedge m = l + \frac{u-l}{2} \wedge \\
& Pre(\textsc{Merge-Sort}(A_0, l, u)) \wedge \\
& Pos(\textsc{Merge-Sort}(A_1, l, m)) \wedge \\
& Pos(\textsc{Merge-Sort}(A_2, m + 1, u)) \wedge \\
& Pos(\textsc{Merge}(A_3, l, m, u)) \\
\Rightarrow & Pos(\textsc{Merge-Sort}(A_3, l, u))
\end{array}
$$

$$
\begin{array}{rl}
\Gamma_1 : & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
\wedge & l \geq u \\
\Rightarrow & A[1 \,..\, l-1] = \langle a_1, \ldots a_{l-1} \rangle \\
\wedge & A[u+1 \,..\, n] = \langle a_{u+1}, \ldots a_n \rangle \wedge \\
\wedge & A[l \,..\, u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\wedge & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{rl}
\Gamma_1 : & A = \langle a_1, \ldots a_n\rangle \land 1 \leq l, u \leq n \\
\land & l \geq u \\
\Rightarrow & A[1\,..\,l-1] = \langle a_1, \ldots a_{l-1}\rangle \\
\land & A[u+1\,..\,n] = \langle a_{u+1}, \ldots a_n\rangle \land \\
\land & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u\rangle) \\
\land & sorted(A, l, u)
\end{array}
$$

$$H \Rightarrow P \land Q \equiv H \Rightarrow P, H \Rightarrow Q$$

PPgSC

$$
\begin{array}{llll}
\Gamma_{1,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & A[1\,..\,l-1] = \langle a_1, \ldots a_{l-1} \rangle \\
\hline
\Gamma_{1,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & A[u+1\,..\,n] = \langle a_{u+1}, \ldots a_n \rangle \\
\hline
\Gamma_{1,3}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{lcl}
\Gamma_{1,1}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & A[1\,..\,l-1] = \langle a_1, \ldots a_{l-1} \rangle \\
\hline
\Gamma_{1,2}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & A[u+1\,..\,n] = \langle a_{u+1}, \ldots a_n \rangle \\
\hline
\Gamma_{1,3}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{lrl}
\Gamma_{1,2}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & A[u+1 \mathinner{..} n] = \langle a_{u+1}, \ldots a_n \rangle \\
\hline
\Gamma_{1,3}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & A[l \mathinner{..} u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4}: & & A = \langle a_1, \ldots a_n \rangle \land 1 \leq l, u \leq n \\
& \land & l \geq u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{rrl}
\Gamma_{1,3}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

PPgSC

$$\frac{\begin{array}{lrl} \Gamma_{1,3}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\ & \wedge & l \geq u \\ & \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \end{array}}{\begin{array}{lrl} \Gamma_{1,4}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\ & \wedge & l \geq u \\ & \Rightarrow & sorted(A, l, u) \end{array}}$$

$l \geq u \Leftrightarrow l = u \vee l > u$

$$
\begin{array}{lrl}
\Gamma_{1,3}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & A[l \,..\, u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l \geq u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$l \geq u \Leftrightarrow l = u \vee l > u$$

$$H \wedge (P_1 \vee P_2) \Rightarrow Q \equiv H \wedge P_1 \Rightarrow Q, H \wedge P_2 \Rightarrow Q$$

$$
\begin{array}{rrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{lrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & A[l\,..\,u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{lrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \le l, u \le n \\
& \wedge & l = u \\
& \Rightarrow & A[l \,..\, l] \in \mathit{permutation}(\langle a_l \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \le l, u \le n \\
& \wedge & l > u \\
& \Rightarrow & A[l \,..\, u] \in \mathit{permutation}(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \le l, u \le n \\
& \wedge & l = u \\
& \Rightarrow & \mathit{sorted}(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \le l, u \le n \\
& \wedge & l > u \\
& \Rightarrow & \mathit{sorted}(A, l, u)
\end{array}
$$

PPgSC

$$
\begin{array}{lrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l = u \\
 & \Rightarrow & A[l \,..\, l] \in permutation(\langle a_l \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l > u \\
 & \Rightarrow & A[l \,..\, u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l = u \\
 & \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l > u \\
 & \Rightarrow & sorted(A, l, u)
\end{array}
$$

PPgSC

$$
\begin{array}{lrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l = u \\
 & \Rightarrow & \langle a_1, \ldots a_n \rangle [l \,..\, l] \in permutation(\langle a_l \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l > u \\
 & \Rightarrow & A[l \,..\, u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l = u \\
 & \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
 & \wedge & l > u \\
 & \Rightarrow & sorted(A, l, u)
\end{array}
$$

PPgSC

$$
\begin{array}{lrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & \langle a_1, \ldots a_n \rangle [l \mathinner{..} l] \in permutation(\langle a_l \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & A[l \mathinner{..} u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{lrl}
\Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & \langle a_l \rangle \in permutation(\langle a_l \rangle) \\
\hline
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & A[l \mathinner{..} u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$\begin{array}{rrl} \Gamma_{1,3,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\ & \wedge & l = u \\ & \Rightarrow & true \\ \hline \Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\ & \wedge & l > u \\ & \Rightarrow & A[l \mathrel{..} u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\ \hline \Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\ & \wedge & l = u \\ & \Rightarrow & sorted(A, l, u) \\ \hline \Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\ & \wedge & l > u \\ & \Rightarrow & sorted(A, l, u) \end{array}$$

$$
\begin{array}{lrl}
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & A[l \mathinner{..} u] \in permutation(\langle a_l, \ldots, a_u \rangle) \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{rll}
\color{red}{\Gamma_{1,3,2}:} & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & \langle\rangle \in permutation(\langle\rangle) \\
\hline
\color{red}{\Gamma_{1,4,1}:} & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\color{red}{\Gamma_{1,4,2}:} & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

PPgSC

$$
\begin{array}{lcl}
\Gamma_{1,3,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & true \\
\hline
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

$$
\begin{array}{rll}
\Gamma_{1,4,1}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l = u \\
& \Rightarrow & sorted(A, l, u) \\
\hline
\Gamma_{1,4,2}: & & A = \langle a_1, \ldots a_n \rangle \wedge 1 \leq l, u \leq n \\
& \wedge & l > u \\
& \Rightarrow & sorted(A, l, u)
\end{array}
$$

idem

# Exercício

1. A ordenação por fusão lança mão de um arranjo auxiliar $B$ na operação de fusão, em cada nível da recursão.
   1.1 Determine a complexidade espacial do algoritmo MERGE-SORT.
   1.2 Descreve as adaptações necessárias para ter um algoritmo similar com complexidade espacial *linear*.
2. Escreva por extenso as obrigações de prova $\Gamma_2$, $\Gamma_3$, $\Gamma_4$, $\Gamma_5$. Verifique se elas são passíveis de serem provadas válidas.

PPgSC