

Aula 6

Divisão e Conquista: Merge-Sort

Projeto e Análise de Algoritmos

Professor Eurinardo Rodrigues Costa
Universidade Federal do Ceará
Campus Russas

2021.1

Sumário

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Aulas Passadas

Algoritmo de Divisão e Conquista

Algoritmo Merge-Sort

Métodos de Resolução de Recorrências

Análise do Tempo do Merge-Sort

Aulas Passadas

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

► Correção de Algoritmos Iterativos

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos
 - ▶ Torre de Hanói

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos
 - ▶ Torre de Hanói
 - ▶ Análise do Tempo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos
 - ▶ Torre de Hanói
 - ▶ Análise do Tempo
 - ▶ Obter uma recursão

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos
 - ▶ Torre de Hanói
 - ▶ Análise do Tempo
 - ▶ Obter uma recursão
 - ▶ Método de substituição:

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos
 - ▶ Torre de Hanói
 - ▶ Análise do Tempo
 - ▶ Obter uma recursão
 - ▶ Método de substituição:
 - Ter um palpite e mostrar por indução

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

- ▶ Correção de Algoritmos Iterativos
 - ▶ Invariante de laço
 - ▶ Exemplo: Insertion-Sort
- ▶ Complexidade de Tempo/Espaço
 - ▶ Pior caso, Melhor caso e Caso médio.
 - ▶ Exemplo: Insertion-Sort
- ▶ Crescimento de Funções
 - ▶ Notação O , Ω e Θ
- ▶ Algoritmos Recursivos
 - ▶ Torre de Hanói
 - ▶ Análise do Tempo
 - ▶ Obter uma recursão
 - ▶ Método de substituição:
 - Ter um palpite e mostrar por indução
 - ▶ Método da árvore de recursão

Algoritmo de Divisão e Conquista

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo de Divisão e Conquista

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Método de Divisão e Conquista

Método de Divisão e Conquista

Fases:

Método de Divisão e Conquista

Fases:

Dividir:

Método de Divisão e Conquista

Fases:

Dividir: dividir o problema em subproblemas menores

Método de Divisão e Conquista

Fases:

Dividir: dividir o problema em subproblemas menores

Conquistar:

Método de Divisão e Conquista

Fases:

Dividir: dividir o problema em subproblemas menores

Conquistar: resolver os subproblemas **recursivamente**

Método de Divisão e Conquista

Fases:

Dividir: dividir o problema em subproblemas menores

Conquistar: resolver os subproblemas **recursivamente**

Combinar:

Método de Divisão e Conquista

Fases:

Dividir: dividir o problema em subproblemas menores

Conquistar: resolver os subproblemas **recursivamente**

Combinar: combinar as subsoluções dos subproblemas em uma solução do problema original.

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

**Algoritmo
Merge-Sort**

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 2: Merge-Sort(A, p, r)

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 3: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 4: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$ **Saída:** $A[p \cdots r]$ ordenado

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 5: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$

Saída: $A[p \cdots r]$ ordenado

1 **se** $p < r$ **então**

|

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 6: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$ **Saída:** $A[p \cdots r]$ ordenado

1 **se** $p < r$ **então**

2 $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$

|

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 7: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$

Saída: $A[p \cdots r]$ ordenado

```
1 se  $p < r$  então
2    $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$ 
3   Merge-Sort( $A, p, q$ )
```

Merge-Sort: ideia e Algoritmo

Algoritmo 8: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$

Saída: $A[p \dots r]$ ordenado

```
1 se  $p < r$  então
2    $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$ 
3   Merge-Sort( $A, p, q$ )
4   Merge-Sort( $A, q + 1, r$ )
```

Merge-Sort: ideia e Algoritmo

Algoritmo 9: Merge-Sort(A, p, r)

Entrada: A : vetor de inteiros e $p \leq r$

Saída: $A[p \dots r]$ ordenado

```
1 se  $p < r$  então
2    $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$ 
3   Merge-Sort( $A, p, q$ )
4   Merge-Sort( $A, q+1, r$ )
5   Intercala( $A, p, q, r$ )
```

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

**Algoritmo
Merge-Sort**

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Algoritmo 11: Intecala(A, p, q, r)

Aulas Passadas

Algoritmo de
Divisão e
Conquista

**Algoritmo
Merge-Sort**

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 12: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Algoritmo 13: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 14: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \dots r]$ ordenado

$$1 \quad n_1 \leftarrow q - p + 1;$$

Merge-Sort: ideia e Algoritmo

Algoritmo 15: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

1 $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$

Merge-Sort: ideia e Algoritmo

PAA - Aula 6

Prof. Eurinardo

Algoritmo 16: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 17: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
- 3 $L[n_1 + 1] \leftarrow \infty;$

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Algoritmo 18: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \dots r]$ ordenado

- ```

1 $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$
2 criar vetor $L[1 \dots n_1 + 1]$ e $R[1 \dots n_2 + 1]$
3 $L[n_1 + 1] \leftarrow \infty; R[n_2 + 1] \leftarrow \infty;$

```

Aulas Passadas

## Algoritmo de Divisão e Conquista

## Algoritmo Merge-Sort

## Métodos de Resolução de Recorrências

## Análise do Tempo do Merge-Sort

---

**Algoritmo 19:** Intecala( $A, p, q, r$ )

---

**Entrada:**  $A[p \cdots q]$  e  $A[q + 1 \cdots r]$ : ordenados

**Saída:**  $A[p \dots r]$  ordenado

- ```

1  $n_1 \leftarrow q - p + 1$ ;  $n_2 \leftarrow r - q$ ;
2 criar vetor  $L[1 \cdots n_1 + 1]$  e  $R[1 \cdots n_2 + 1]$ 
3  $L[n_1 + 1] \leftarrow \infty$ ;  $R[n_2 + 1] \leftarrow \infty$ ;
4 para  $i \leftarrow 1$  até  $n_1$  faça

```

Aulas Passadas

Algoritmo de Divisão e Conquista

Algoritmo Merge-Sort

Métodos de Resolução de Recorrências

Análise do Tempo do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 20: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1$; $n_2 \leftarrow r - q$;
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
- 3 $L[n_1 + 1] \leftarrow \infty$; $R[n_2 + 1] \leftarrow \infty$;
- 4 **para** $i \leftarrow 1$ **até** n_1 **faça**
- 5 $L[i] \leftarrow A[p + i - 1]$

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 21: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1$; $n_2 \leftarrow r - q$;
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
- 3 $L[n_1 + 1] \leftarrow \infty$; $R[n_2 + 1] \leftarrow \infty$;
- 4 **para** $i \leftarrow 1$ **até** n_1 **faça**
- 5 $L[i] \leftarrow A[p + i - 1]$
- 6 **para** $i \leftarrow 1$ **até** n_2 **faça**
 |

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 22: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1$; $n_2 \leftarrow r - q$;
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
- 3 $L[n_1 + 1] \leftarrow \infty$; $R[n_2 + 1] \leftarrow \infty$;
- 4 **para** $i \leftarrow 1$ **até** n_1 **faça**
- 5 $L[i] \leftarrow A[p + i - 1]$
- 6 **para** $i \leftarrow 1$ **até** n_2 **faça**
- 7 $R[i] \leftarrow A[q + i]$

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 23: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1$; $n_2 \leftarrow r - q$;
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
- 3 $L[n_1 + 1] \leftarrow \infty$; $R[n_2 + 1] \leftarrow \infty$;
- 4 **para** $i \leftarrow 1$ **até** n_1 **faça**
- 5 $L[i] \leftarrow A[p + i - 1]$
- 6 **para** $i \leftarrow 1$ **até** n_2 **faça**
- 7 $R[i] \leftarrow A[q + i]$
- 8 $i \leftarrow 1$;

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 24: Intecala(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$
- 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
- 3 $L[n_1 + 1] \leftarrow \infty; R[n_2 + 1] \leftarrow \infty;$
- 4 **para** $i \leftarrow 1$ **até** n_1 **faça**
- 5 $L[i] \leftarrow A[p + i - 1]$
- 6 **para** $i \leftarrow 1$ **até** n_2 **faça**
- 7 $R[i] \leftarrow A[q + i]$
- 8 $i \leftarrow 1; j \leftarrow 1;$

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 25: Intecal(A, p, q, r)

Entrada: $A[p \cdots q]$ e $A[q + 1 \cdots r]$: ordenados

Saída: $A[p \cdots r]$ ordenado

- 1 $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$
 - 2 **criar vetor** $L[1 \cdots n_1 + 1]$ e $R[1 \cdots n_2 + 1]$
 - 3 $L[n_1 + 1] \leftarrow \infty; R[n_2 + 1] \leftarrow \infty;$
 - 4 **para** $i \leftarrow 1$ **até** n_1 **faça**
 - 5 $L[i] \leftarrow A[p + i - 1]$
 - 6 **para** $i \leftarrow 1$ **até** n_2 **faça**
 - 7 $R[i] \leftarrow A[q + i]$
 - 8 $i \leftarrow 1; j \leftarrow 1;$
 - 9 **para** $k \leftarrow p$ **até** r **faça**
-

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 26: Intecal(A, p, q, r)

Entrada: $A[p \dots q]$ e $A[q + 1 \dots r]$: ordenados

Saída: $A[p \dots r]$ ordenado

```
1  $n_1 \leftarrow q - p + 1$ ;  $n_2 \leftarrow r - q$ ;  
2 criar vetor  $L[1 \dots n_1 + 1]$  e  $R[1 \dots n_2 + 1]$   
3  $L[n_1 + 1] \leftarrow \infty$ ;  $R[n_2 + 1] \leftarrow \infty$ ;  
4 para  $i \leftarrow 1$  até  $n_1$  faça  
5    $L[i] \leftarrow A[p + i - 1]$   
6 para  $i \leftarrow 1$  até  $n_2$  faça  
7    $R[i] \leftarrow A[q + i]$   
8  $i \leftarrow 1$ ;  $j \leftarrow 1$ ;  
9 para  $k \leftarrow p$  até  $r$  faça  
10   se  $L[i] < R[j]$  então  
      |  
      |
```

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 27: Intecal(A, p, q, r)

Entrada: $A[p \dots q]$ e $A[q + 1 \dots r]$: ordenados

Saída: $A[p \dots r]$ ordenado

```
1  $n_1 \leftarrow q - p + 1$ ;  $n_2 \leftarrow r - q$ ;  
2 criar vetor  $L[1 \dots n_1 + 1]$  e  $R[1 \dots n_2 + 1]$   
3  $L[n_1 + 1] \leftarrow \infty$ ;  $R[n_2 + 1] \leftarrow \infty$ ;  
4 para  $i \leftarrow 1$  até  $n_1$  faça  
5    $L[i] \leftarrow A[p + i - 1]$   
6 para  $i \leftarrow 1$  até  $n_2$  faça  
7    $R[i] \leftarrow A[q + i]$   
8  $i \leftarrow 1$ ;  $j \leftarrow 1$ ;  
9 para  $k \leftarrow p$  até  $r$  faça  
10   se  $L[i] < R[j]$  então  
11      $A[k] \leftarrow L[i]$ 
```

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 28: Intecal(A, p, q, r)

Entrada: $A[p \dots q]$ e $A[q + 1 \dots r]$: ordenados

Saída: $A[p \dots r]$ ordenado

```
1  $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$   
2 criar vetor  $L[1 \dots n_1 + 1]$  e  $R[1 \dots n_2 + 1]$   
3  $L[n_1 + 1] \leftarrow \infty; R[n_2 + 1] \leftarrow \infty;$   
4 para  $i \leftarrow 1$  até  $n_1$  faça  
5    $L[i] \leftarrow A[p + i - 1]$   
6 para  $i \leftarrow 1$  até  $n_2$  faça  
7    $R[i] \leftarrow A[q + i]$   
8  $i \leftarrow 1; j \leftarrow 1;$   
9 para  $k \leftarrow p$  até  $r$  faça  
10   se  $L[i] < R[j]$  então  
11      $A[k] \leftarrow L[i]$   
12      $i \leftarrow i + 1$ 
```

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 29: Intecal(A, p, q, r)

Entrada: $A[p \dots q]$ e $A[q + 1 \dots r]$: ordenados

Saída: $A[p \dots r]$ ordenado

```
1  $n_1 \leftarrow q - p + 1; n_2 \leftarrow r - q;$ 
2 criar vetor  $L[1 \dots n_1 + 1]$  e  $R[1 \dots n_2 + 1]$ 
3  $L[n_1 + 1] \leftarrow \infty; R[n_2 + 1] \leftarrow \infty;$ 
4 para  $i \leftarrow 1$  até  $n_1$  faça
5    $L[i] \leftarrow A[p + i - 1]$ 
6 para  $i \leftarrow 1$  até  $n_2$  faça
7    $R[i] \leftarrow A[q + i]$ 
8  $i \leftarrow 1; j \leftarrow 1;$ 
9 para  $k \leftarrow p$  até  $r$  faça
10   se  $L[i] < R[j]$  então
11      $A[k] \leftarrow L[i]$ 
12      $i \leftarrow i + 1$ 
13   senão
```

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Merge-Sort: ideia e Algoritmo

Algoritmo 30: Intecal(A, p, q, r)**Entrada:** $A[p \dots q]$ e $A[q + 1 \dots r]$: ordenados**Saída:** $A[p \dots r]$ ordenado

```
1  $n_1 \leftarrow q - p + 1$ ;  $n_2 \leftarrow r - q$ ;  
2 criar vetor  $L[1 \dots n_1 + 1]$  e  $R[1 \dots n_2 + 1]$   
3  $L[n_1 + 1] \leftarrow \infty$ ;  $R[n_2 + 1] \leftarrow \infty$ ;  
4 para  $i \leftarrow 1$  até  $n_1$  faça  
5    $L[i] \leftarrow A[p + i - 1]$   
6 para  $i \leftarrow 1$  até  $n_2$  faça  
7    $R[i] \leftarrow A[q + i]$   
8  $i \leftarrow 1$ ;  $j \leftarrow 1$ ;  
9 para  $k \leftarrow p$  até  $r$  faça  
10   se  $L[i] < R[j]$  então  
11      $A[k] \leftarrow L[i]$   
12      $i \leftarrow i + 1$   
13   senão  
14      $A[k] \leftarrow R[j]$ 
```

Merge-Sort: ideia e Algoritmo

Algoritmo 31: Intercala(A, p, q, r)**Entrada:** $A[p \dots q]$ e $A[q + 1 \dots r]$: ordenados**Saída:** $A[p \dots r]$ ordenado

```
1  $n_1 \leftarrow q - p + 1$ ;  $n_2 \leftarrow r - q$ ;  
2 criar vetor  $L[1 \dots n_1 + 1]$  e  $R[1 \dots n_2 + 1]$   
3  $L[n_1 + 1] \leftarrow \infty$ ;  $R[n_2 + 1] \leftarrow \infty$ ;  
4 para  $i \leftarrow 1$  até  $n_1$  faça  
5    $L[i] \leftarrow A[p + i - 1]$   
6 para  $i \leftarrow 1$  até  $n_2$  faça  
7    $R[i] \leftarrow A[q + i]$   
8  $i \leftarrow 1$ ;  $j \leftarrow 1$ ;  
9 para  $k \leftarrow p$  até  $r$  faça  
10   se  $L[i] < R[j]$  então  
11      $A[k] \leftarrow L[i]$   
12      $i \leftarrow i + 1$   
13   senão  
14      $A[k] \leftarrow R[j]$   
15      $j \leftarrow j + 1$ 
```

Resolução de Recorrências

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Resolução de Recorrências

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort

Método de Substituição

Método de Substituição

1. Arriscar um palpite em forma de solução

Método de Substituição

1. Arriscar um palpite em forma de solução
2. Usar indução para mostrar que funciona.

Método de Substituição

1. Arriscar um palpite em forma de solução
2. Usar indução para mostrar que funciona.

Método da Árvore de Recurção

Método de Substituição

1. Arriscar um palpite em forma de solução
2. Usar indução para mostrar que funciona.

Método da Árvore de Recurção

1. Construir a árvore de recursão.

Método de Substituição

1. Arriscar um palpite em forma de solução
2. Usar indução para mostrar que funciona.

Método da Árvore de Recurção

1. Construir a árvore de recursão.
2. Calcular o custo de cada nível da árvore.

Método de Substituição

1. Arriscar um palpite em forma de solução
2. Usar indução para mostrar que funciona.

Método da Árvore de Recurção

1. Construir a árvore de recursão.
2. Calcular o custo de cada nível da árvore.
3. Tempo = soma dos custos em cada nível.

Análise do Tempo do Merge-Sort

PAA - Aula 6

Prof. Eurinardo

Aulas Passadas

Algoritmo de
Divisão e
Conquista

Algoritmo
Merge-Sort

Métodos de
Resolução de
Recorrências

Análise do Tempo
do Merge-Sort



LEISERSON, C.E., STEIN, C., RIVEST, R.L.,
CORMEN T.H.

Algoritmos: teoria e prática, 3ed.

Editora Campus, ano 2012.

Obrigado!