

Estrutura de Dados I

Merge-Sort

Prof. Rodrigo Minetto

rminetto@dainf.ct.utfpr.edu.br

Universidade Tecnológica Federal do Paraná

Material compilado de: Cormen.

Sumário

- 1 Introdução
- 2 Algoritmo
- 3 Execução
- 4 Considerações

Merge-Sort

O algoritmo Merge-Sort foi inventado em 1945 por Von Neumann. O algoritmo segue o paradigma de divisão e conquista (*divide and conquer*):

- Dividir: descubra o ponto médio do sub-arranjo (tempo constante).
- Conquistar: resolva recursivamente dois sub-problemas de tamanho $n/2$.
- Combinar: combine os dois sub-arranjos em um único conjunto ordenado (tempo de n).

Sumário

- 1 Introdução
- 2 Algoritmo**
- 3 Execução
- 4 Considerações

Merge-Sort

Merge-Sort (A , e , d)

1. se $e < d$ então
2. $m \leftarrow \lfloor (e + d) / 2 \rfloor$;
3. Merge-Sort (A , e , m);
4. Merge-Sort (A , $m + 1$, d);
5. Intercala (A , e , m , d);

e

m

d

5	2	4	7	1	3	2	6
---	---	---	---	---	---	---	---

Sumário

- 1 Introdução
- 2 Algoritmo
- 3 Execução
- 4 Considerações

Execução

5	2	4	7	1	3	2	6
---	---	---	---	---	---	---	---

Execução

5	2	4	7	1	3	2	6
---	---	---	---	---	---	---	---

5	2	4	7
---	---	---	---

1	3	2	6
---	---	---	---

Execução

5	2	4	7	1	3	2	6
---	---	---	---	---	---	---	---

5	2	4	7
---	---	---	---

1	3	2	6
---	---	---	---

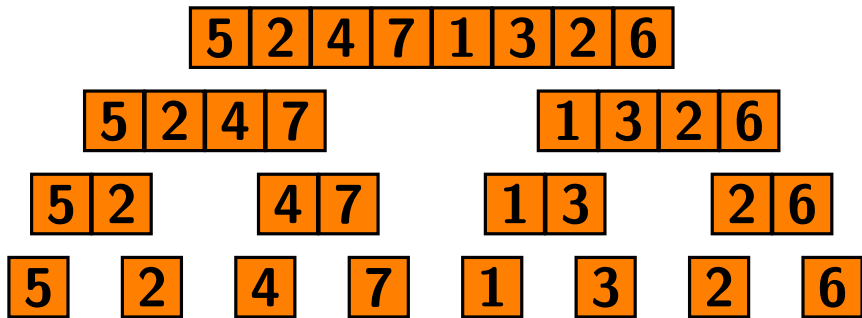
5	2
---	---

4	7
---	---

1	3
---	---

2	6
---	---

Execução



Execução

5 2 4 7 1 3 2 6

5 2 4 7

1 3 2 6

5 2

4 7

1 3

2 6

5

2

4

7

1

3

2

6

2 5

4 7

1 3

2 6

Execução

5 2 4 7 1 3 2 6

5 2 4 7

1 3 2 6

5 2

4 7

1 3

2 6

5

2

4

7

1

3

2

6

2 5

4 7

1 3

2 6

2 4 5 7

1 2 3 6

Execução

5 2 4 7 1 3 2 6

5 2 4 7

1 3 2 6

5 2

4 7

1 3

2 6

5

2

4

7

1

3

2

6

2 5

4 7

1 3

2 6

2 4 5 7

1 2 3 6

1 2 2 3 4 5 6 7

Sumário

- 1 Introdução
- 2 Algoritmo
- 3 Execução
- 4 Considerações**

Complexidade

O algoritmo do Merge-Sort é estável e sua equação de recorrência é dado por:

$$T(\mathbf{n}) = \begin{cases} \Theta(1) & \text{se } \mathbf{n} = 1, \\ T(\lceil \mathbf{n}/2 \rceil) + T(\lfloor \mathbf{n}/2 \rfloor) + \Theta(\mathbf{n}) & \text{se } \mathbf{n} > 1. \end{cases}$$

Complexidade de tempo: $\Theta(n \log n)$.

Complexidade de espaço: $\Theta(n)$.