

Aula 9

Heap-Sort

Projeto e Análise de Algoritmos

Professor Eurinardo Rodrigues Costa
Universidade Federal do Ceará
Campus Russas

2021.1

Sumário

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

► Algoritmos de Ordenação

- ▶ Algoritmos de Ordenação
 - ▶ Insertion-Sort,

- ▶ Algoritmos de Ordenação
 - ▶ Insertion-Sort,
 - ▶ Merge-Sort e

- ▶ Algoritmos de Ordenação
 - ▶ Insertion-Sort,
 - ▶ Merge-Sort e
 - ▶ Quick-Sort.

Heap Máximo

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

Heap Máximo

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

Heap Máximo

Heap Máximo

Tabela T que pode vista como uma árvore binária com as seguintes propriedades:

Heap Máximo

Tabela T que pode vista como uma árvore binária com as seguintes propriedades:

► $T[i] \geq T[2i]$

Heap Máximo

Tabela T que pode vista como uma árvore binária com as seguintes propriedades:

- ▶ $T[i] \geq T[2i]$
- ▶ $T[i] \geq T[2i + 1]$

Heap Máximo

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

Algoritmo 2: Max-Heapify(A, i)

Algoritmo 3: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Algoritmo 4: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos**Saída:** $A[i]$ Heap Máximo

Algoritmo 5: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos**Saída:** $A[i]$ Heap Máximo

1 $n \leftarrow$ tamanho de A

Algoritmo 6: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i;$

Algoritmo 7: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$

Algoritmo 8: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$
- 3 **se** $\ell \leq n$ **e** $A[\ell] > A[i]$ **então**
 |

Algoritmo 9: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$
- 3 **se** $\ell \leq n$ **e** $A[\ell] > A[i]$ **então**
- 4 | maior $\leftarrow \ell$

Algoritmo 10: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$
- 3 **se** $\ell \leq n$ **e** $A[\ell] > A[i]$ **então**
- 4 | maior $\leftarrow \ell$
- 5 **senão**

Algoritmo 11: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$
- 3 **se** $\ell \leq n$ **e** $A[\ell] > A[i]$ **então**
- 4 | maior $\leftarrow \ell$
- 5 **senão** maior $\leftarrow i$

Algoritmo 12: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos

Saída: $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$
- 3 **se** $\ell \leq n$ **e** $A[\ell] > A[i]$ **então**
- 4 | maior $\leftarrow \ell$
- 5 **senão** maior $\leftarrow i;$
- 6 **se** $r \leq n$ **e** $A[r] > A[\text{maior}]$ **então**
- |

Algoritmo 13: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos**Saída:** $A[i]$ Heap Máximo

- 1 $n \leftarrow$ tamanho de A
- 2 $\ell \leftarrow 2i; r \leftarrow 2i + 1;$
- 3 **se** $\ell \leq n$ **e** $A[\ell] > A[i]$ **então**
- 4 $\text{maior} \leftarrow \ell$
- 5 **senão** $\text{maior} \leftarrow i;$
- 6 **se** $r \leq n$ **e** $A[r] > A[\text{maior}]$ **então**
- 7 $\text{maior} \leftarrow r$

Algoritmo 14: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos**Saída:** $A[i]$ Heap Máximo

```
1  $n \leftarrow$  tamanho de  $A$ 
2  $\ell \leftarrow 2i; r \leftarrow 2i + 1;$ 
3 se  $\ell \leq n$  e  $A[\ell] > A[i]$  então
4   | maior  $\leftarrow \ell$ 
5 senão maior  $\leftarrow i;$ 
6 se  $r \leq n$  e  $A[r] > A[\text{maior}]$  então
7   | maior  $\leftarrow r$ 
8 se maior  $\neq i$  então
   |
```

Algoritmo 15: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos**Saída:** $A[i]$ Heap Máximo

```
1  $n \leftarrow$  tamanho de  $A$ 
2  $\ell \leftarrow 2i; r \leftarrow 2i + 1;$ 
3 se  $\ell \leq n$  e  $A[\ell] > A[i]$  então
4   |   maior  $\leftarrow \ell$ 
5 senão maior  $\leftarrow i;$ 
6 se  $r \leq n$  e  $A[r] > A[\text{maior}]$  então
7   |   maior  $\leftarrow r$ 
8 se maior  $\neq i$  então
9   |   trocar  $A[i] \leftrightarrow A[\text{maior}]$ 
   |
```

Algoritmo 16: Max-Heapify(A, i)

Entrada: $A[2i]$ e $A[2i + 1]$: são Heaps Máximos**Saída:** $A[i]$ Heap Máximo

```
1   $n \leftarrow$  tamanho de  $A$ 
2   $\ell \leftarrow 2i; r \leftarrow 2i + 1;$ 
3  se  $\ell \leq n$  e  $A[\ell] > A[i]$  então
4    |   maior  $\leftarrow \ell$ 
5  senão maior  $\leftarrow i;$ 
6  se  $r \leq n$  e  $A[r] > A[\text{maior}]$  então
7    |   maior  $\leftarrow r$ 
8  se maior  $\neq i$  então
9    |   trocar  $A[i] \leftrightarrow A[\text{maior}]$ 
10   |   Max-Heapify( $A, \text{maior}$ )
```

Heap Máximo

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

Algoritmo 18: Build-Max-Heap(A)

Algoritmo 19: Build-Max-Heap(A)

Entrada: A : vetor

Heap Máximo

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

Algoritmo 20: Build-Max-Heap(A)

Entrada: A : vetor**Saída:** $A[1]$ Heap Máximo

Algoritmo 21: Build-Max-Heap(A)

Entrada: A : vetor

Saída: $A[1]$ Heap Máximo

1 $n \leftarrow$ tamanho de A

Algoritmo 22: Build-Max-Heap(A)

Entrada: A : vetor

Saída: $A[1]$ Heap Máximo

- ```
1 $n \leftarrow$ tamanho de A
2 para $i \leftarrow \lfloor \frac{n}{2} \rfloor$ até 1 faça
3 \quad Max-Heapify(A, i)
```
-

# Heap Máximo

EDA - Aula 9

Prof. Eurinardo

Aulas Passadas

Heap Máximo

Algoritmos

Max-Heapify

Build-Max-Heap

Heap-Sort

---

## Algoritmo 24: Heap-Sort( $A$ )

---

---

## Algoritmo 25: Heap-Sort( $A$ )

---

**Entrada:**  $A$ : vetor

---

## **Algoritmo 26:** Heap-Sort( $A$ )

---

**Entrada:**  $A$ : vetor

**Saída:**  $A$  ordenado

---

## Algoritmo 27: Heap-Sort( $A$ )

---

**Entrada:**  $A$ : vetor

**Saída:**  $A$  ordenado

1 Build-Max-Heap( $A$ )

---

**Algoritmo 28: Heap-Sort( $A$ )**

---

**Entrada:**  $A$ : vetor

**Saída:**  $A$  ordenado

- 1 Build-Max-Heap( $A$ )
  - 2 **para**  $i \leftarrow \text{tam}(A)$  **até** 2 **faça**
- |



---

**Algoritmo 29: Heap-Sort( $A$ )**

---

**Entrada:**  $A$ : vetor

**Saída:**  $A$  ordenado

```
1 Build-Max-Heap(A)
2 para $i \leftarrow \text{tam}(A)$ até 2 faça
3 | trocar $A[1] \leftrightarrow A[i]$
```

---

**Algoritmo 30: Heap-Sort( $A$ )**

---

**Entrada:**  $A$ : vetor

**Saída:**  $A$  ordenado

```
1 Build-Max-Heap(A)
2 para $i \leftarrow \text{tam}(A)$ até 2 faça
3 | trocar $A[1] \leftrightarrow A[i]$
4 | $\text{tam}(A) \leftarrow \text{tam}(A) - 1$
```

---

**Algoritmo 31: Heap-Sort( $A$ )**

---

**Entrada:**  $A$ : vetor

**Saída:**  $A$  ordenado

```
1 Build-Max-Heap(A)
2 para $i \leftarrow \text{tam}(A)$ até 2 faça
3 | trocar $A[1] \leftrightarrow A[i]$
4 | $\text{tam}(A) \leftarrow \text{tam}(A) - 1$
5 | Max-Heapify(A , 1)
```

---



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

*Algoritmos: teoria e prática, 3ed.*

Editora Campus, ano 2012.

# Obrigado!