

# ORDENAÇÃO: *HEAPSORT*

PUC MINAS

ENGENHARIA DE SOFTWARE

# HEAPSORT

- Problema do método de ordenação

## **Seleção:**

- custo para se selecionar o elemento que ocupará a posição de referência.
- E se...
  - a seleção puder ser feita a baixo custo?

# HEAP

- Estrutura de dados que **não** garante a ordem FIFO;
  - mas sim a ordem de prioridade pré-estabelecida;
  - o primeiro que sai;
    - é o elemento que apresenta a maior prioridade.

# HEAP

- Em geral, implementado em um vetor ou em uma árvore, com as seguintes regras:
  - $A[i] \geq A[2 * i];$
  - $A[i] \geq A[2 * i + 1].$
- Exemplo:
  - $A[2]$  é maior ou igual a  $A[4]$  e  $A[5]$ .

# HEAP

- Operações:
  - construir o *heap* em um vetor;
  - refazer o *heap* a cada inserção ou retirada.

# CONSTRUINDO O *HEAP*

23	42	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

23	42	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

23	42	8	15	16	4	95
1	2	3	4	5	6	7



# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7



# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	8	15	16	4	95
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	95	15	16	4	8
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	95	15	16	4	8
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

42	23	95	15	16	4	8
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

95	23	42	15	16	4	8
1	2	3	4	5	6	7




# CONSTRUINDO O *HEAP*

95	23	42	15	16	4	8
1	2	3	4	5	6	7

# CONSTRUINDO O *HEAP*

95	23	42	15	16	4	8
1	2	3	4	5	6	7



# ALGORITMO *HEAPSORT*

- Construir o *heap*;
- Para cada posição de referência, a partir do fim do conjunto:
  - trocar o elemento que ocupa a posição de referência com o primeiro;
  - refazer o *heap* a partir da primeira posição.

# ORDENAÇÃO COM *HEAPSORT*

95	42	23	15	16	4	8
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

95	42	23	15	16	4	8
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

<b>95</b>	<b>42</b>	<b>23</b>	<b>15</b>	<b>16</b>	<b>4</b>	<b>8</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>

# ORDENAÇÃO COM *HEAPSORT*

8	42	23	15	16	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

8	42	23	15	16	4	95
1	2	3	4	5	6	7



# ORDENAÇÃO COM *HEAPSORT*

42	8	23	15	16	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

42	8	23	15	16	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

42	8	23	15	16	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

42	16	23	15	8	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

42	16	23	15	8	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

42	16	23	15	8	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

42	16	23	15	8	4	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

<b>4</b>	<b>16</b>	<b>23</b>	<b>15</b>	<b>8</b>	<b>42</b>	<b>95</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>



# ORDENAÇÃO COM *HEAPSORT*

4	16	23	15	8	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

23	16	4	15	8	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

23	16	4	15	8	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

23	16	4	15	8	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

<b>23</b>	<b>16</b>	<b>4</b>	<b>15</b>	<b>8</b>	<b>42</b>	<b>95</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>

# ORDENAÇÃO COM *HEAPSORT*

8	16	4	15	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

8	16	4	15	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

16	8	4	15	23	42	95
1	2	3	4	5	6	7



# ORDENAÇÃO COM *HEAPSORT*

16	8	4	15	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

16	8	4	15	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

16	15	4	8	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

16	15	4	8	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

16	15	4	8	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

<b>16</b>	<b>15</b>	<b>4</b>	<b>8</b>	<b>23</b>	<b>42</b>	<b>95</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>

# ORDENAÇÃO COM *HEAPSORT*

8	15	4	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

8	15	4	16	23	42	95
1	2	3	4	5	6	7



# ORDENAÇÃO COM *HEAPSORT*

15	8	4	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

15	8	4	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

15	8	4	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

15	8	4	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

4	8	15	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

4	8	15	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

8	4	15	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

8	4	15	16	23	42	95
1	2	3	4	5	6	7



# ORDENAÇÃO COM *HEAPSORT*

8	4	15	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

8	4	15	16	23	42	95
1	2	3	4	5	6	7

# ORDENAÇÃO COM *HEAPSORT*

<b>4</b>	<b>8</b>	<b>15</b>	<b>16</b>	<b>23</b>	<b>42</b>	<b>95</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>

# ORDENAÇÃO COM *HEAPSORT*

4	8	15	16	23	42	95
1	2	3	4	5	6	7

# CONSIDERAÇÕES

- Método não-estável.
- Custo alto para construir o *heap*.