

Algoritmos e Estruturas de Dados II

Heapsort

Prof. Flávio José M. Coelho

fcoelho@uea.edu.br

Objetivos

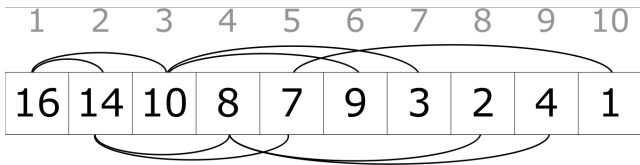
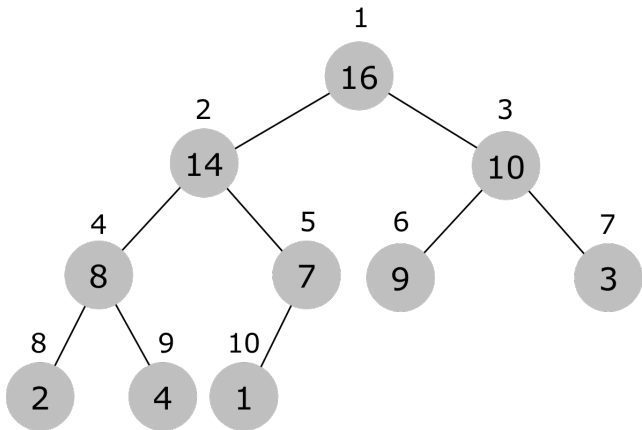
Entender o funcionamento da estrutura de dados **Heap**

Entender o funcionamento do algoritmo de ordenação **Heapsort**

Heap

Heapsort

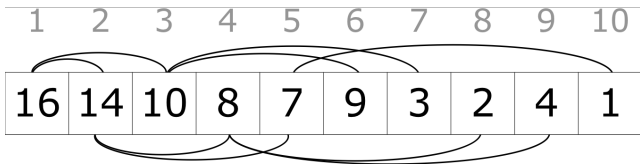
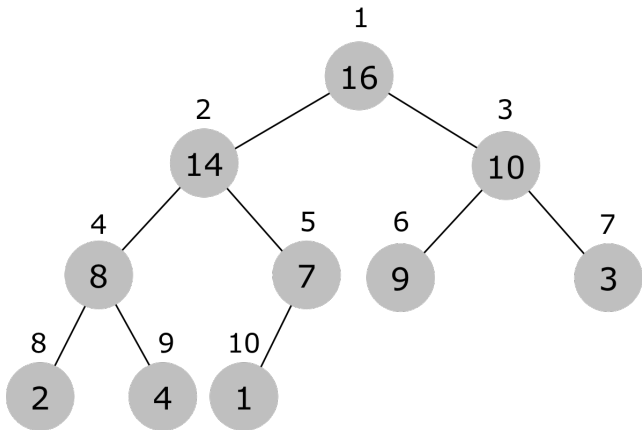
Um **heap** (binário) é um vetor usado para representar uma árvore binária quase completa.



Heap

Heap máximo (*max-heap*). Chave de cada nó é maior ou igual às chaves de seus filhos.

Heap mínimo (*min-heap*). Chave de cada nó é menor ou igual às chaves de seus filhos.



Heap

Seja $A[1..n]$ um **heap binário** (máximo).

- ▶ $A.tamanho = n$ é o tamanho do vetor A .
- ▶ $A.tamHeap$ é o tamanho do heap (número de itens válidos no heap, contidos em A).
- ▶ $0 \leq A.tamHeap \leq A.tamanho$.

Heap

Seja $A[1..n]$ um **heap binário** (máximo).

- ▶ $A[1]$ é a raiz da árvore.
- ▶ Dado um índice i , obtém-se o pai de i e os filhos de i por meio dos procedimentos $\text{PAI}(i)$, $\text{ESQ}(i)$ E $\text{DIR}(i)$.

PAI(i)

1 **retorne** $\lfloor i/2 \rfloor$

ESQ(i)

1 **retorne** $2i$

DIR(i)

1 **retorne** $2i + 1$

Heap

Seja $A[1..n]$ um **heap binário** (máximo).

- ▶ A **propriedade do heap máximo** deve ser satisfeita para todo nó i , não-raiz:
 $A[\text{PAI}(i)] \geq A[i]$.

Heap

Seja $A[1..n]$ um **heap binário** (máximo).

- ▶ Os procedimentos $\text{HEAPFICA}(A, i)$ e $\text{CONSTROI-HEAP}(A)$ mantêm a propriedade do heap e constrói o heap, respectivamente.

HEAPFICA(A, i)

1 $l = \text{ESQ}(i), r = \text{DIR}(i)$

3 **se** $l \leq A.\text{tamHeap}$ **e** $A[l] > A[i]$

4 $\text{maior} = l$

5 **senão** $\text{maior} = i$

6 **se** $r \leq A.\text{tamHeap}$ **e** $A[r] > A[\text{maior}]$

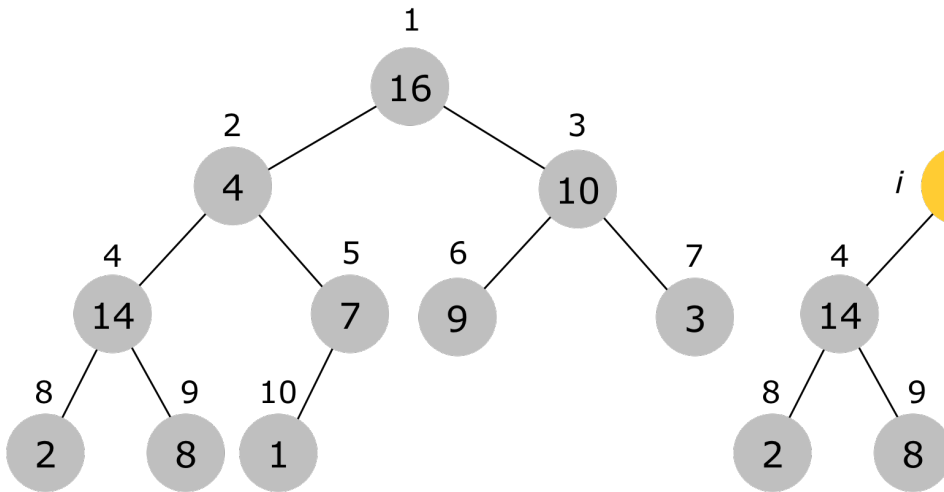
7 $\text{maior} = r$

8 **se** $\text{maior} \neq i$

9 troque $A[i]$ com $A[\text{maior}]$

10 HEAPFICA(A, maior)

Chamada: $\text{HEAPFICA}(A, 2)$



CONSTROI-HEAP(A)

```
1   $A.tamHeap = A.tamanho$   
2  for  $i = \lfloor A.tamanho/2 \rfloor$  até 1  
3      HEAPFICA( $A, i$ )
```


Heapsort

Heapsort



Robert W. Floyd



J. W. J. Williams

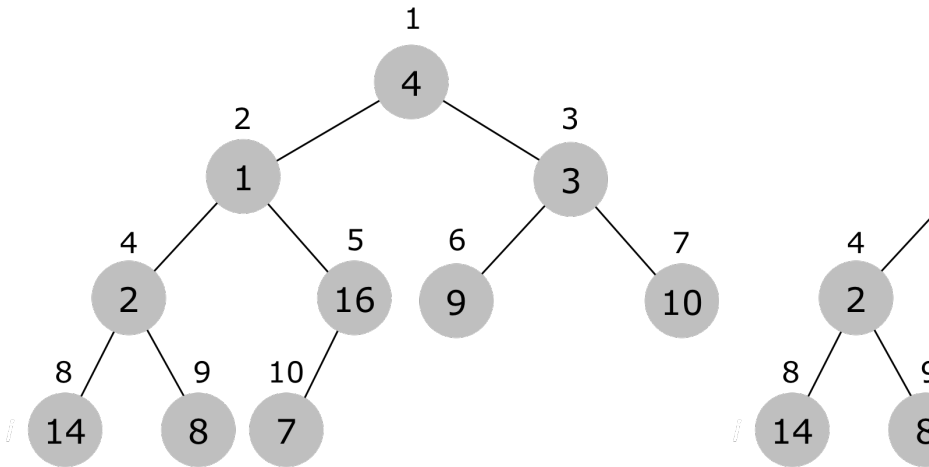
Heapsort foi inventado por Robert W. Floyd e J.W.J Williams nos anos 1960.

Heapsort





O algoritmo usa o heap máximo para obter sempre o maior elemento do vetor (raiz) e coloca-lo na última, penúltima, etc, até a segunda posição do vetor.

HEAPSORT(A)

```
1  CONSTROI-HEAP( $A$ )  
2  for  $i = A.tamanho$  até 2  
3      troque  $A[1]$  com  $A[i]$   
4       $A.tamHeap = A.tamHeap - 1$   
5      HEAPFICA( $A, 1$ )
```



Referências

-  T. H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd edition, MIT Press, 2010
-  A. Levitin. Introduction to the Design and Analysis of Algorithms. 3rd edition. Addison-Wesley, 2007
-  R. Sedgewick, K. Wayne. Algorithms. 4th edition, Addison-Wesley Professional, 2011
-  N. Ziviani. Projeto de Algoritmos com Implementação em Pascal C. Cengage Learning, 2012

Onde obter este material:

`est.uea.edu.br/fcoelho`