

# **Aula 3**

## **Listas de Prioridades**

### **Heaps**

# Introdução

- Dados que possuam prioridade
- A prioridade do dado por variar ao longo do tempo
- Possível selecionar o dado de maior prioridade

# Introdução

- Lista de prioridades
  - Tabela em que cada dado tem uma prioridade associada.
    - Em geral, a prioridade é um valor numérico
  - Operações básicas
    - Seleção do item de maior prioridade
    - Inserção de um novo dado
    - Remoção do dado de maior prioridade
    - Alteração da prioridade de um dado

# Introdução

- Métodos usados
  - Lista não ordenada
  - Lista ordenada
  - Heap

# Complexidade 1

- Lista de prioridade → Lista não ordenada com  $n$  elementos
  - Complexidade:
    - Seleção
    - Inserção
    - Remoção
    - Alteração
    - Construção

# Complexidade 1

- Lista de prioridade → Lista não ordenada com  $n$  elementos
  - Complexidade:
    - Seleção  $O(n)$
    - Inserção  $O(1)$
    - Remoção  $O(n)$
    - Alteração  $O(n)$
    - Construção  $O(n)$

# Complexidade 2

- Lista de prioridade → Lista ordenada com  $n$  elementos
  - Complexidade:
    - Seleção
    - Inserção
    - Remoção
    - Alteração
    - Construção

# Complexidade 2

- Lista de prioridade → Lista ordenada com  $n$  elementos
  - Complexidade:
    - Seleção  $O(1)$
    - Inserção  $O(n)$
    - Remoção  $O(1)$
    - Alteração  $O(n)$
    - Construção  $O(n \log n)$

# Heap

- Lista linear composta com chaves (prioridades) satisfazendo:

$$s_i \leq s_{\lfloor i/2 \rfloor}, 1 \leq i \leq n.$$

- Exemplo:

95 60 78 39 28 66 70 33

# Exercício

- Verifique se as seguintes listas constituem heaps

(i) 33 32 28 31 26 29 25 30 27

(ii) 33 32 28 31 29 26 25 30 27

# Exercício

- Verifique se as seguintes listas constituem heaps

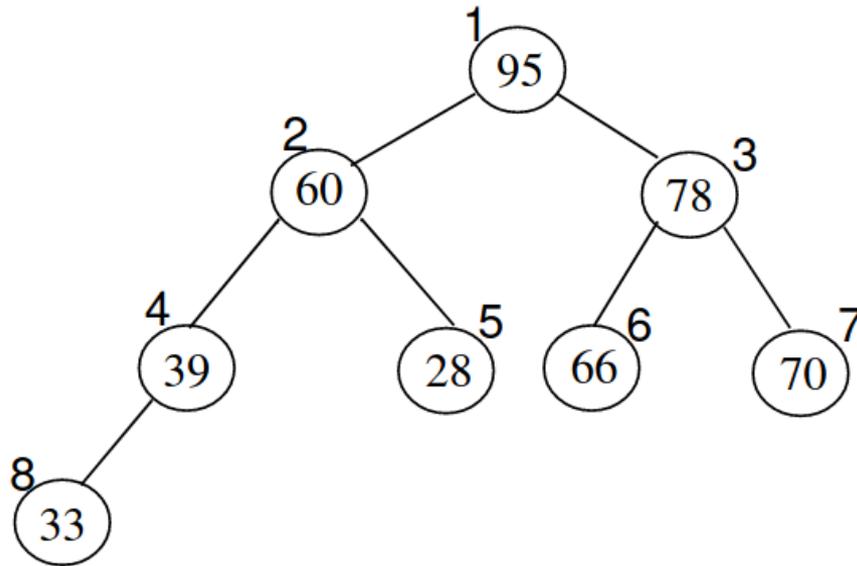
(i)	33	32	28	31	26	29	25	30	27	
(ii)	33	32	28	31	29	26	25	30	27	
	1	2	3	4	5	6	7	8	9	

# Heaps e Árvores

- Um heap pode ser visualizado por uma árvore binária
  - Os nós da árvore são numerados sequencialmente
  - Cada nó representa uma chave e o rótulo a prioridade
  - Cada nó possui a prioridade maior ou igual aos filhos

# Exemplo

i	1	2	3	4	5	6	7	8
s <sub>i</sub>	95	60	78	39	28	66	70	33



# Complexidade 3

- Lista de prioridade → Heap
  - Complexidade:
    - Seleção
    - Inserção
    - Remoção
    - Alteração
    - Construção

# Complexidade 3

- Lista de prioridade → Heap
  - Complexidade:
    - Seleção  $O(1)$
    - Inserção  $O(\log n)$
    - Remoção  $O(\log n)$
    - Alteração  $O(\log n)$
    - Construção  $O(n)$

# Exercício

- Desenhe a árvore binária equivalente ao heap:

33 32 28 31 26 24 25 30 27