Heaps

Filas de Prioridades - Introdução

Heap

Alteração de Prioridade

Inserção / Remoção

Construção

Filas de Prioridades - Introdução

TAD : Lista linear em que se associa a cada elemento um valor inteiro (prioridade) que determina a ordem na qual o elemento será manipulado ("deixa o conjunto")

TAD
- Operações: Seleção / Inserção / Remoção / Alteração / Construção / ...
- Estr.Dados: ?

Operação	**Lista Desordenada**	**Lista Ordenada**	**HEAP**
Seleção	O(n)	O(1)	O(1)
Inserção	O(1)	O(n)	O(log n)
Remoção	O(n)	O(1)	O(log n)
Alteração	O(n)	O(n)	O(log n)
Construção	O(n)	O(n log n)	O(n)

Heap

Estrutura de Dados : Heap

- Heap é definido como uma sequência de elementos com prioridades

$$Pr_1, Pr_2, \ldots, Pr_n$$

tal que

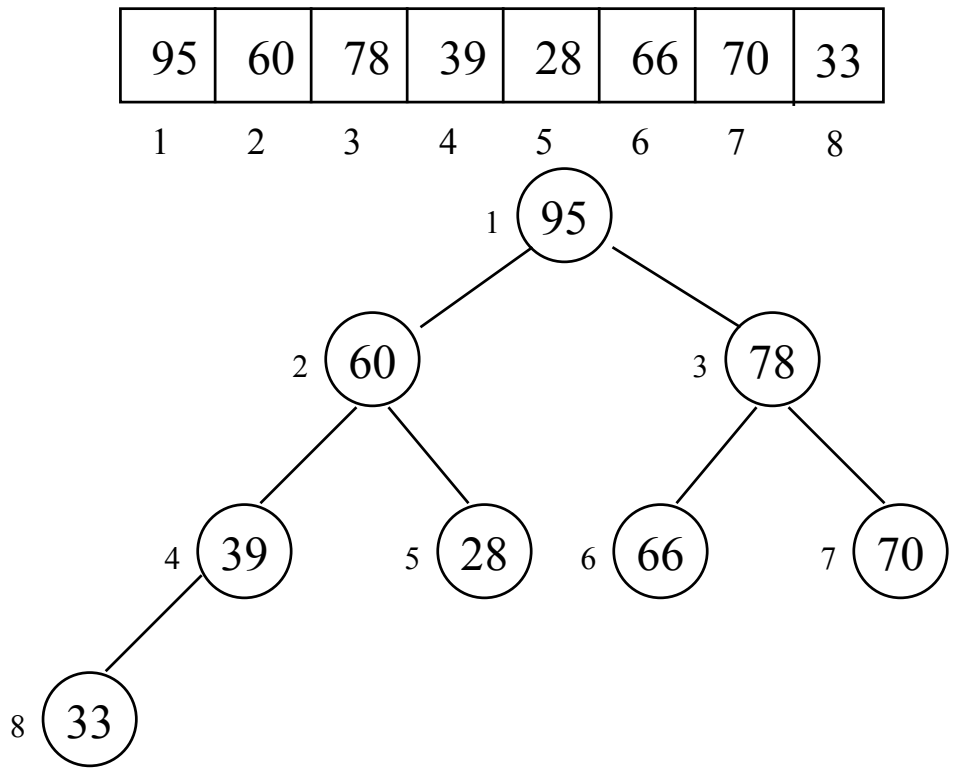
$Pr_i \leq Pr_{\lfloor i/2 \rfloor}$, para $1 < i \leq n$

ou

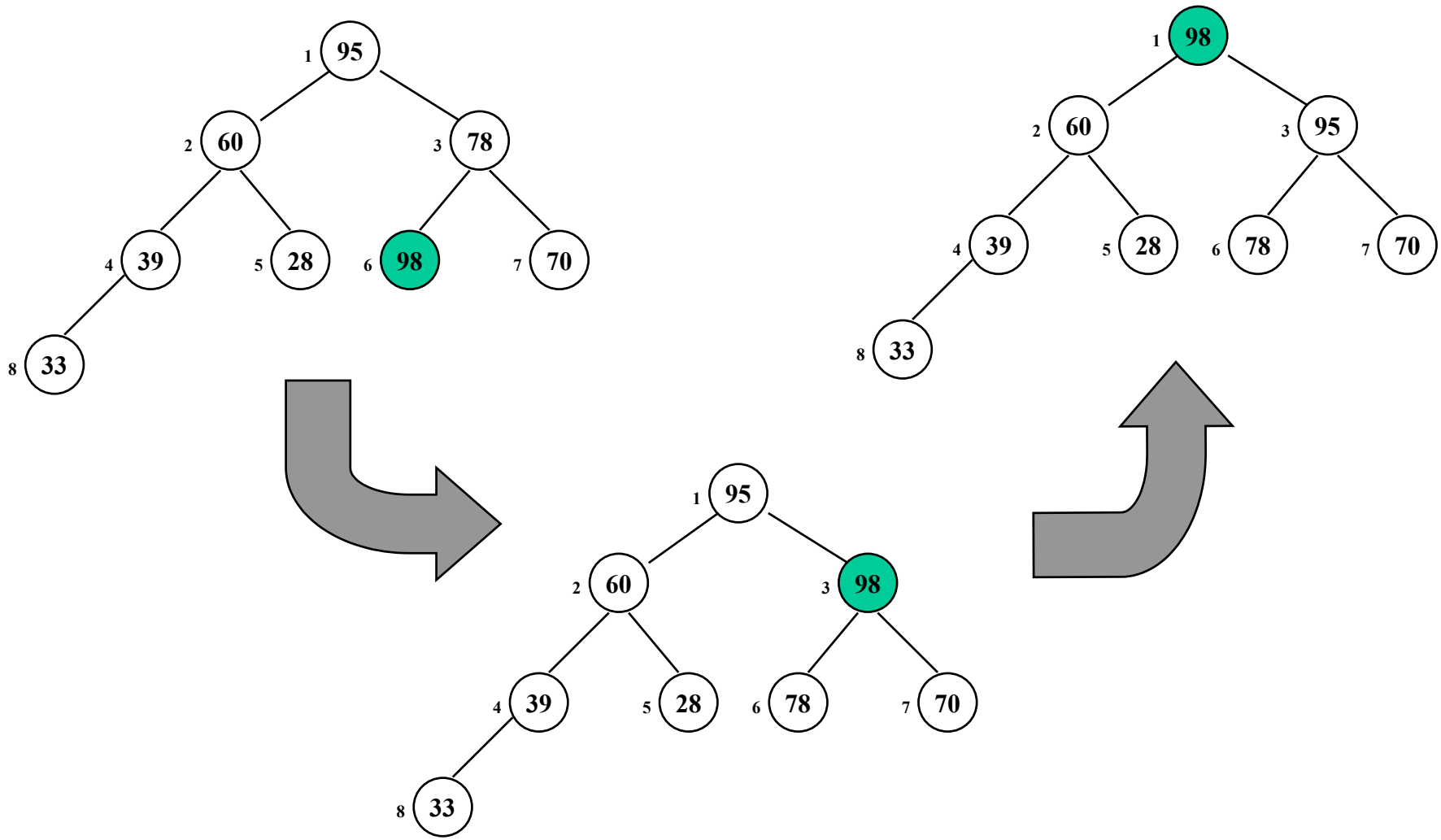
$$\begin{cases} Pr_i \geq Pr_{2i} \\ Pr_i \geq Pr_{2i+1} \end{cases}, \text{ para } 1 \leq i \leq \lfloor n/2 \rfloor$$

Heap pode ser visto como uma árvore binária

Exemplo



Alteração de Prioridade - Aumento



Alteração de Prioridade - Aumento

Procedimento Subir (i) // O(log n)

$pai := \lfloor i / 2 \rfloor$

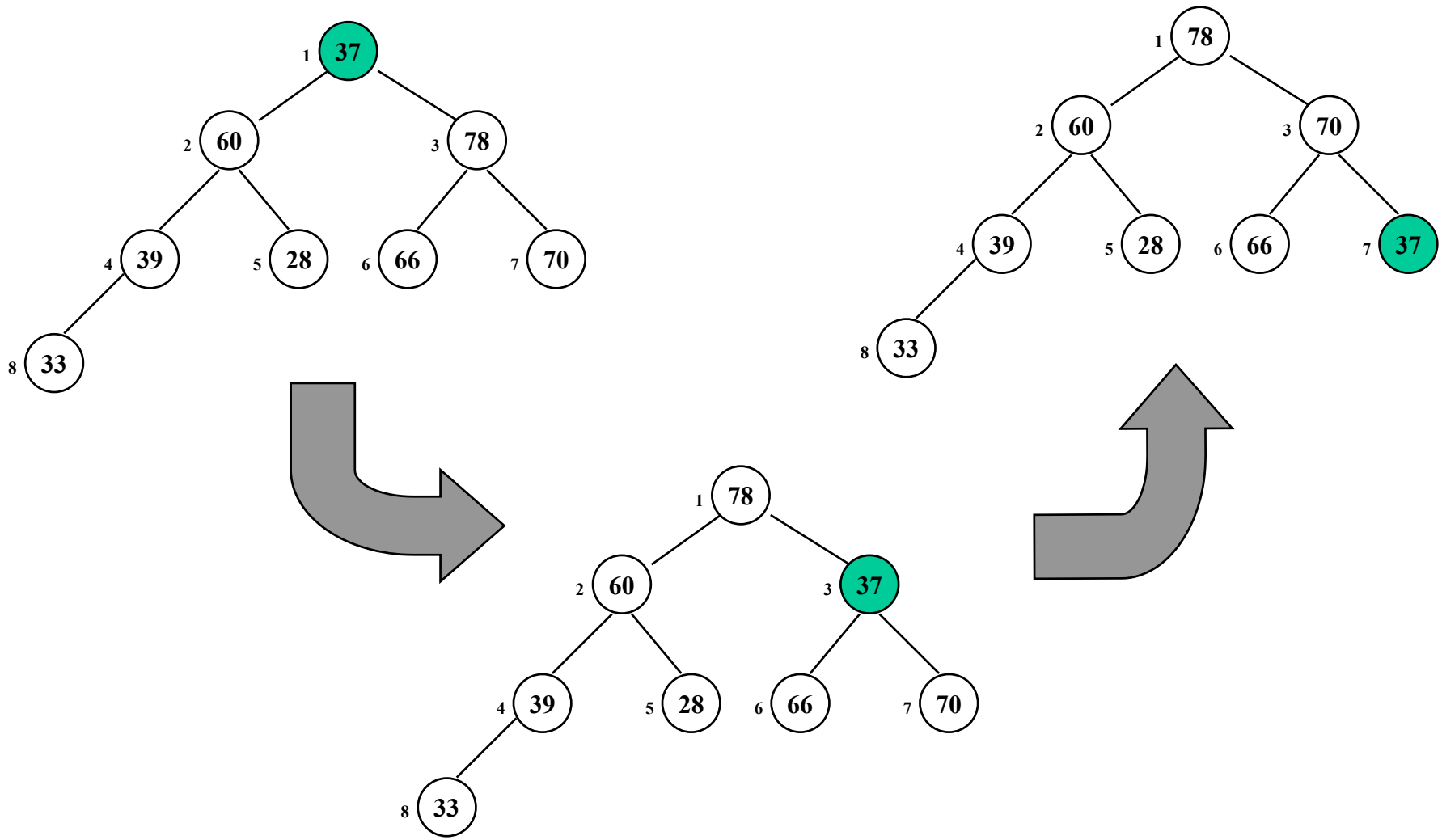
se $pai \geq 1$ então

se $Pr_i > Pr_{pai}$ então

Troca (i, pai)

Subir (pai)

Alteração de Prioridade - Redução



Alteração de Prioridade - Redução

Procedimento Descer (i, n) // O(log n)

$filho := 2 * i$

se $filho \leq n$ então

se $(filho < n)$ e $(Pr_{filho+1} > Pr_{filho})$ então

$filho{+}{+}$

se $Pr_i < Pr_{filho}$ então

Troca (i, $filho$)

Descer ($filho$, n)

Inserção no Heap

Inserção

se $n < tamanhoMaximo$ então // Existe espaço ?

armazena novo elemento na posição $n + 1$

$n := n + 1$

Subir (n)

CUSTO $\rightarrow$ O(log n)

Remoção do Heap

Remoção

se $n > 0$ então // Existe elemento ?

retira elemento da posição 1

substitui elemento da posição 1 pelo n-ésimo

$n := n - 1$

Descer (1, n)

CUSTO $\rightarrow$ O(log n)

Construção do Heap

Versão 1

para $i := 2$ até n faça

Subir (i)

Versão 2

para $i := \lfloor n / 2 \rfloor$ até 1 faça

Descer (i, n)

CUSTO $\rightarrow$ O(n)