



Heap Sort

Diogo Oliveira
diogoespinharaoliveira@gmail.com



A ideia é usar uma pilha de prioridades implementada com um heap.

Heap é uma árvore binária onde um nó possui no máximo dois filhos, e o valor do nó pai é maior do que os filhos.

Intuição

- ▷ Manipular o vetor de forma a representar uma árvore binária;
- ▷ Filhos de um nó i estão nas posições
 - Filho esquerdo: $2 * i + 1$
 - Filho direito: $2 * i + 2$
- ▷ Pai de um nó i está na posição
 - $(i - 1) / 2$
- ▷ *A raiz deve possuir o maior valor;*

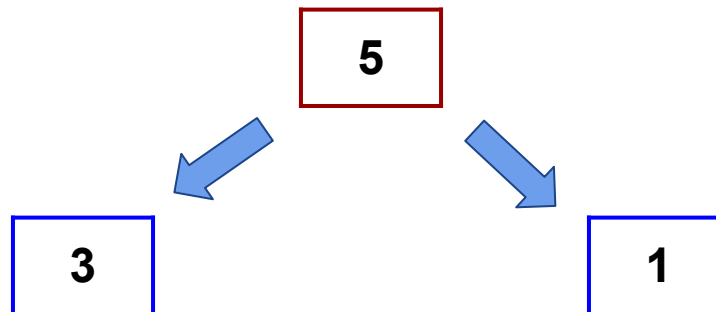
Exemplo

5	3	1	7	6	2	4
0	1	2	3	4	5	6

Exemplo

5	3	1	7	6	2	4
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

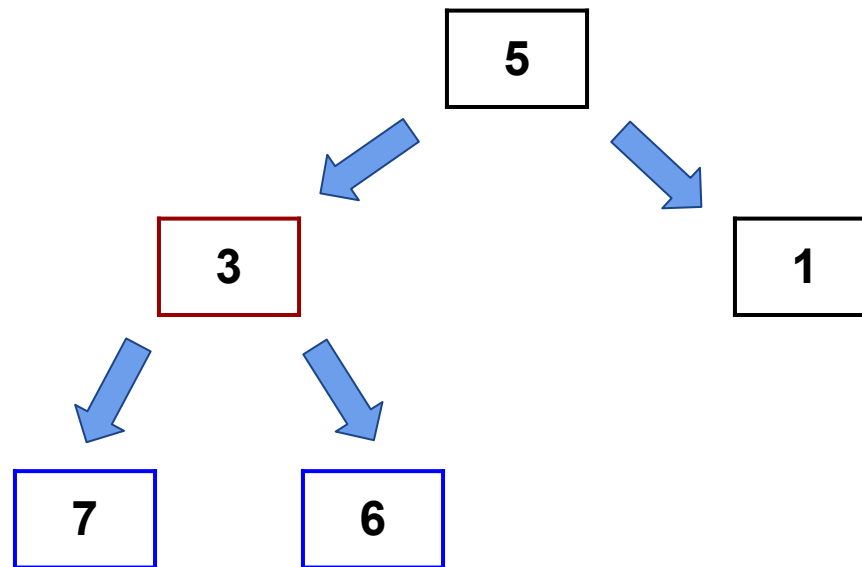
Árvore



Exemplo

5	3	1	7	6	2	4
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

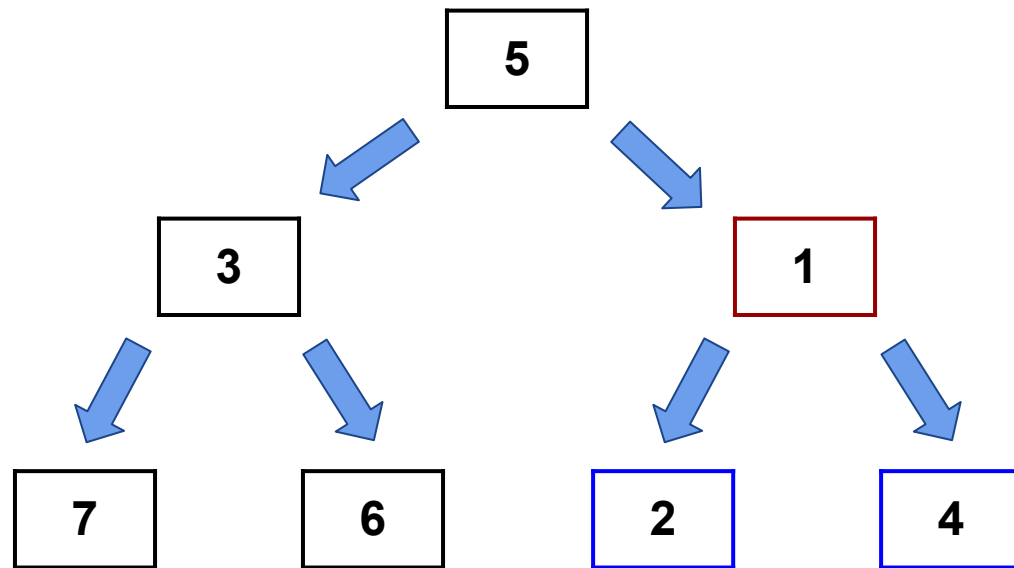
Árvore



Exemplo

5	3	1	7	6	2	4
0	1	2	3	4	5	6
		i			$2i + 1$	$2i + 2$

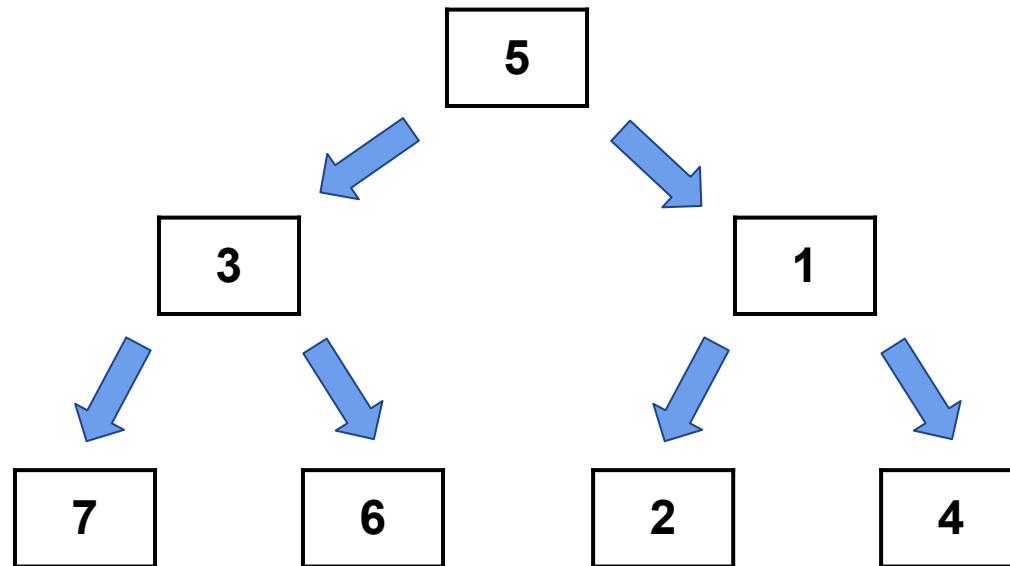
Árvore



Exemplo

5	3	1	7	6	2	4
0	1	2	3	4	5	6

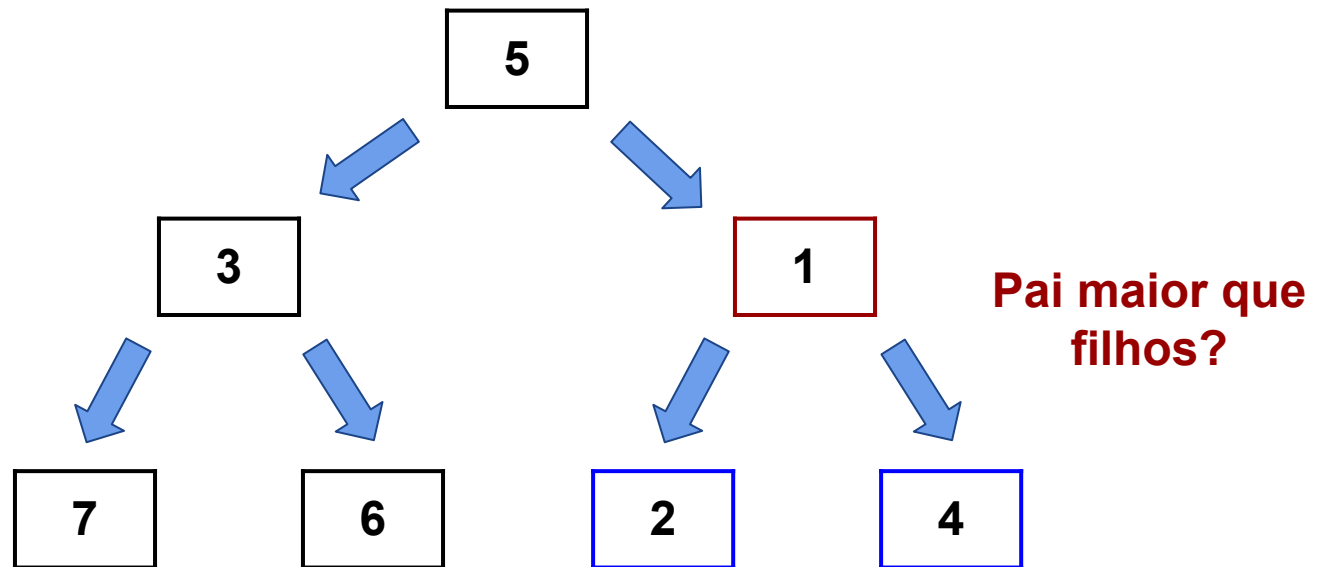
Árvore



Criação do Heap

5	3	1	7	6	2	4
0	1	2	3	4	5	6
i			$2i + 1$		$2i + 2$	

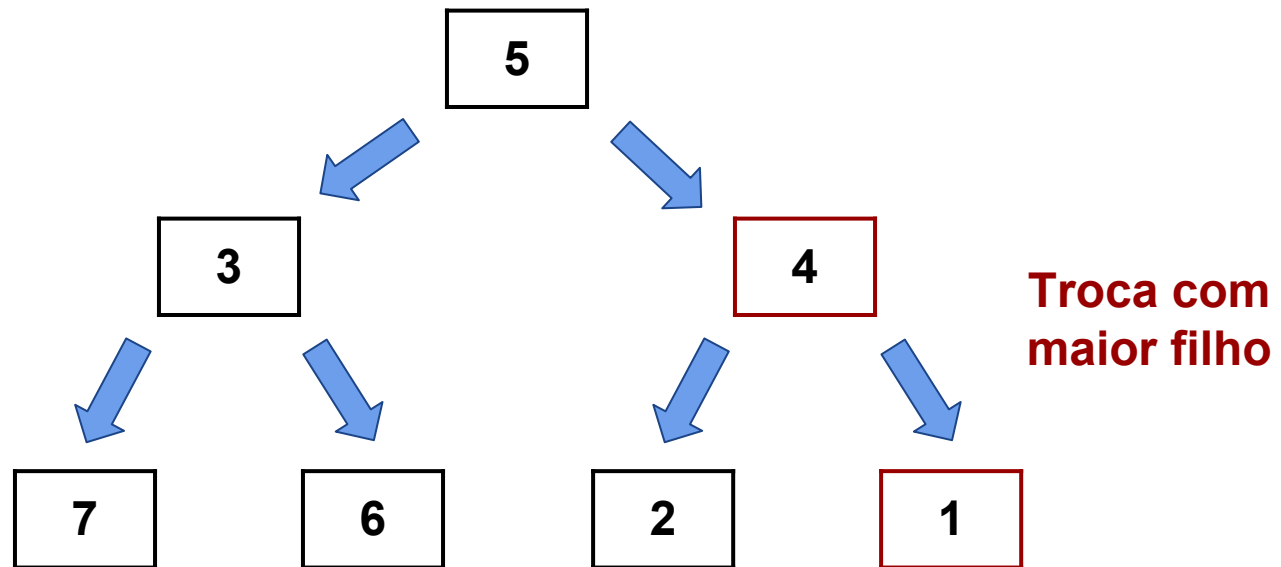
Árvore



Criação do Heap

5	3	4	7	6	2	1
0	1	2	3	4	5	6
i			$2i + 1$		$2i + 2$	

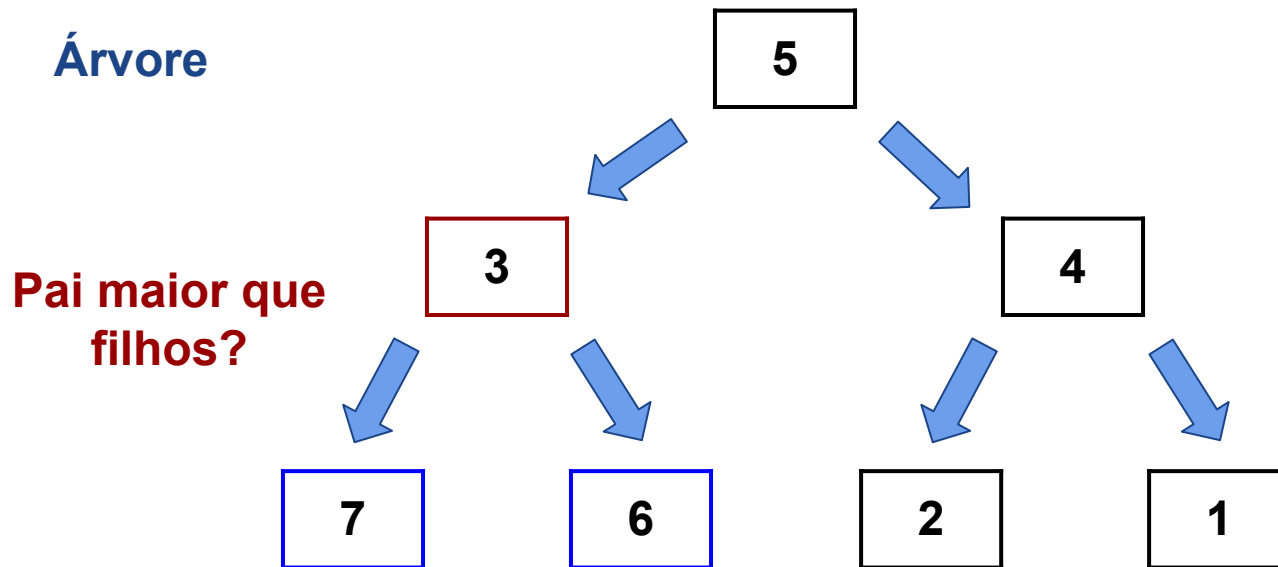
Árvore



Criação do Heap

5	3	4	7	6	2	1
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

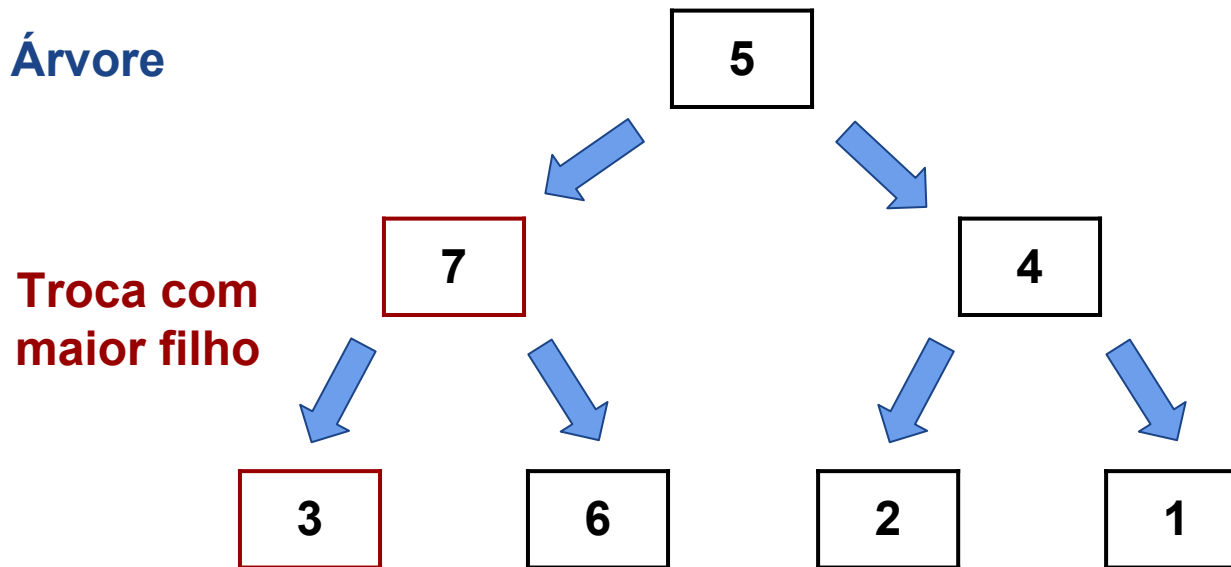
Árvore



Criação do Heap

5	7	4	3	6	2	1
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

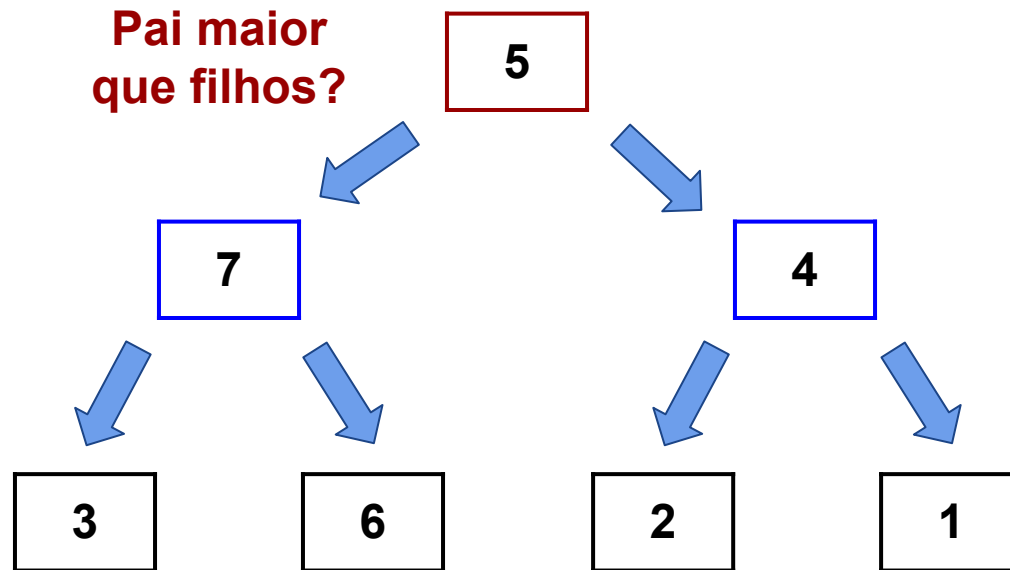
Árvore



Criação do Heap

5	7	4	3	6	2	1
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

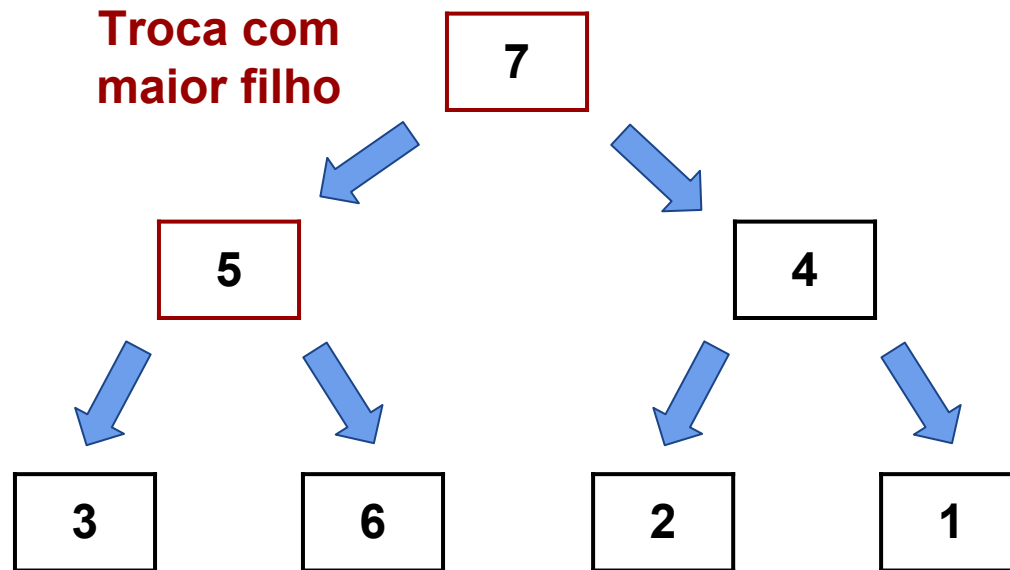
Árvore



Criação do Heap

7	5	4	3	6	2	1
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

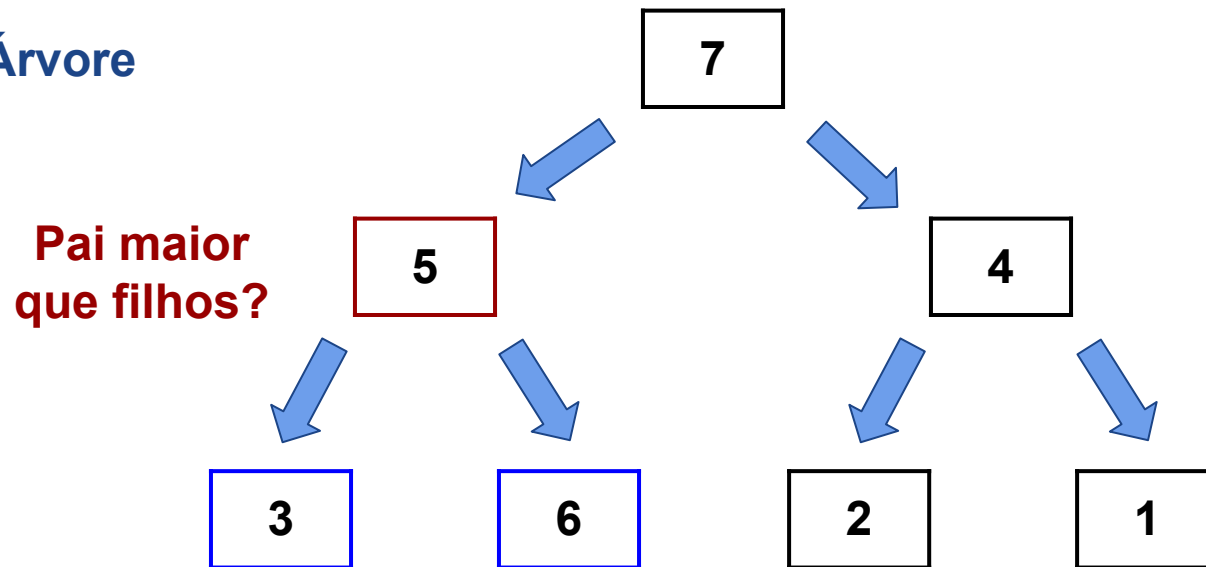
Árvore



Criação do Heap

7	5	4	3	6	2	1
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

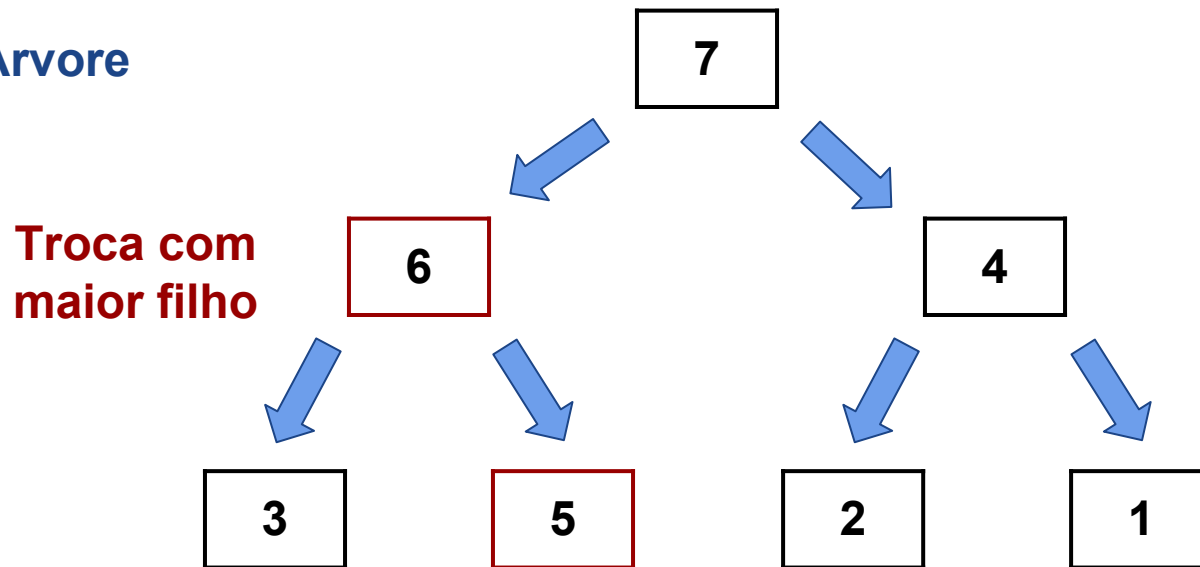
Árvore



Criação do Heap

7	6	4	3	5	2	1
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

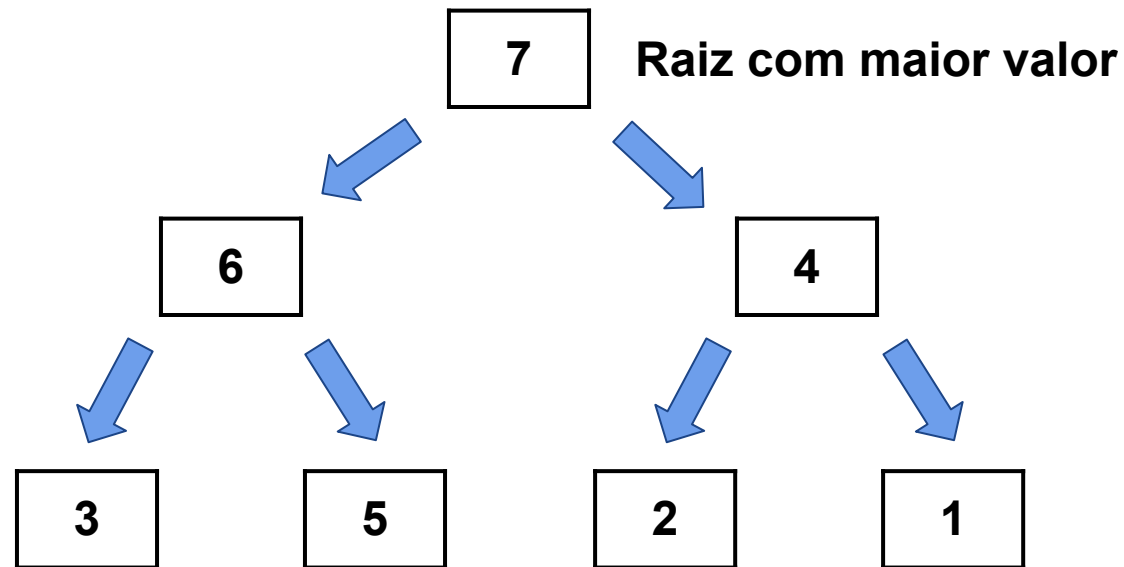
Árvore



Heap criado

7	6	4	3	5	2	1
0	1	2	3	4	5	6

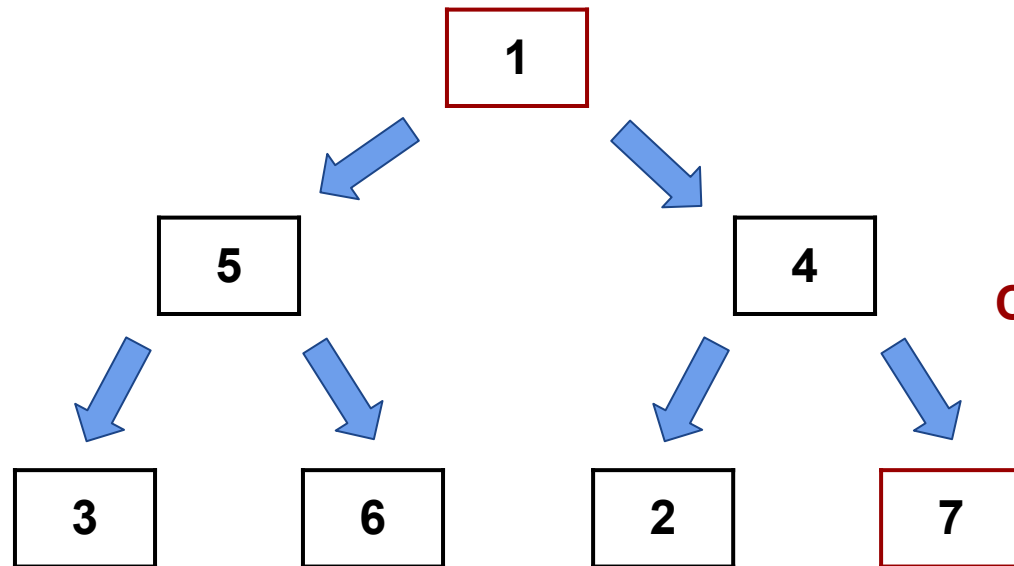
Árvore



Iteração 1

1	6	4	3	5	2	7
0	1	2	3	4	5	6

Árvore

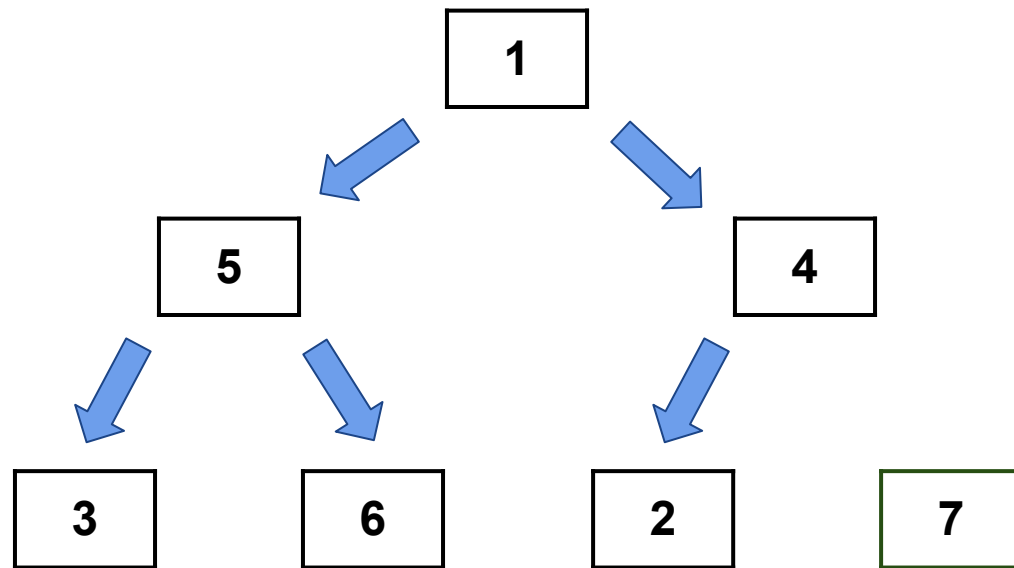


Coloca a raiz no final do vetor

Iteração 1

1	5	4	3	6	2	7
0	1	2	3	4	5	6

Árvore

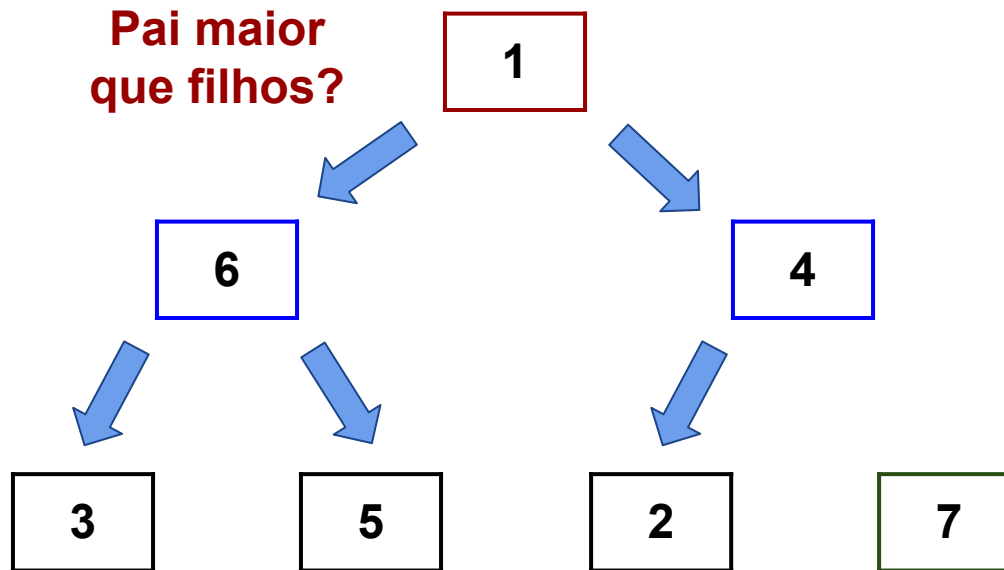


Já está na
posição certa

Iteração 1

1	6	4	3	5	2	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

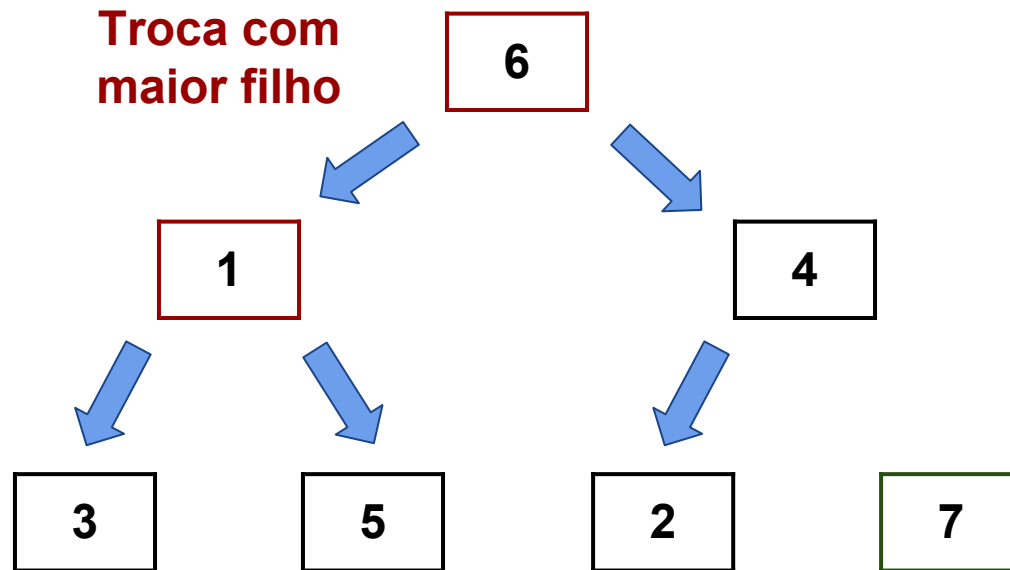
Árvore



Iteração 1

6	1	4	3	5	2	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

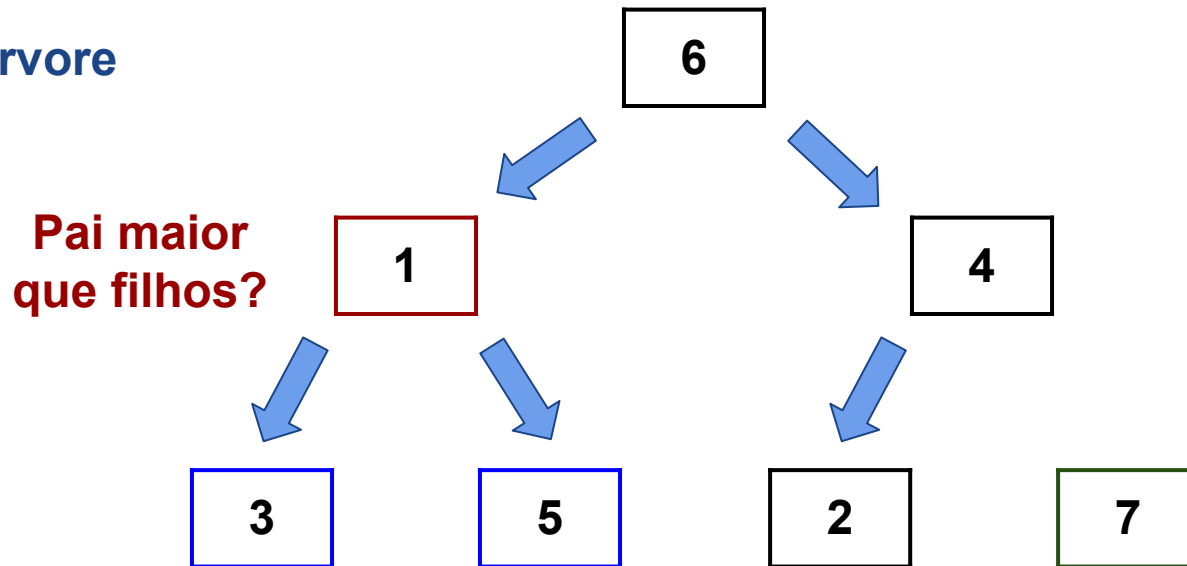
Árvore



Iteração 1

6	1	4	3	5	2	7
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

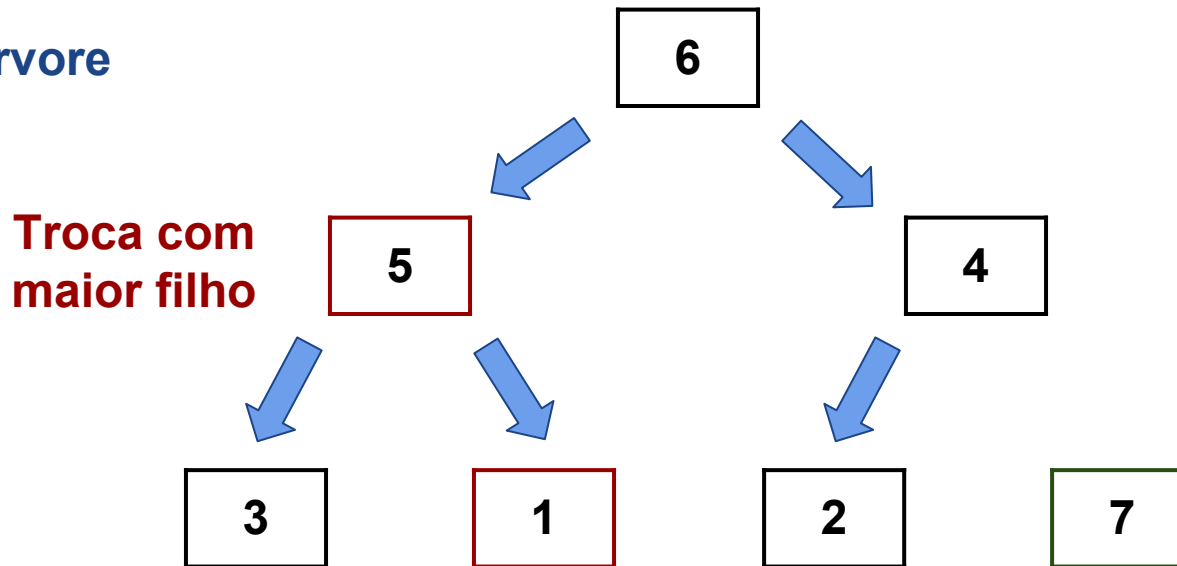
Árvore



Iteração 1

6	5	4	3	1	2	7
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

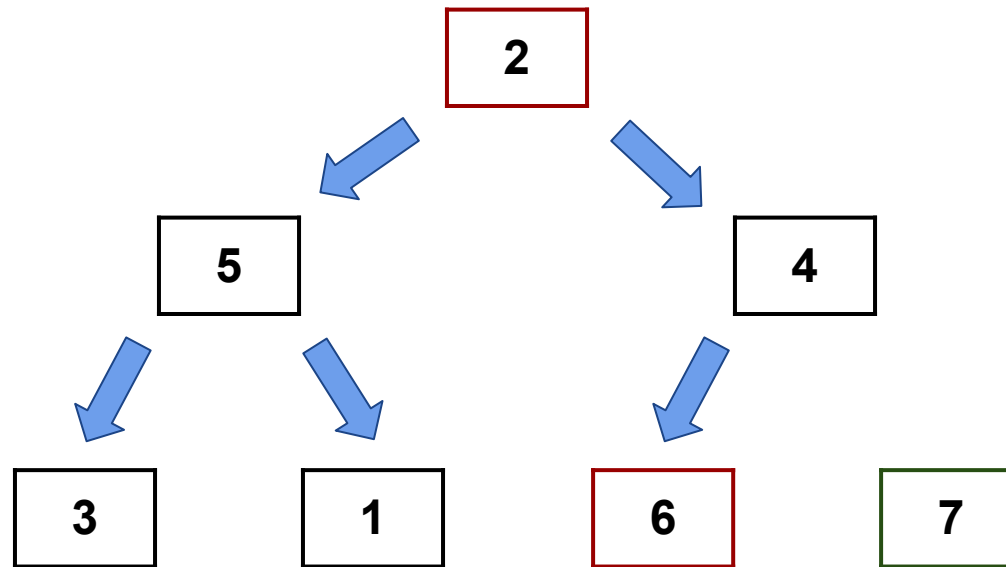
Árvore



Iteração 2

2	5	4	3	1	6	7
0	1	2	3	4	5	6

Árvore

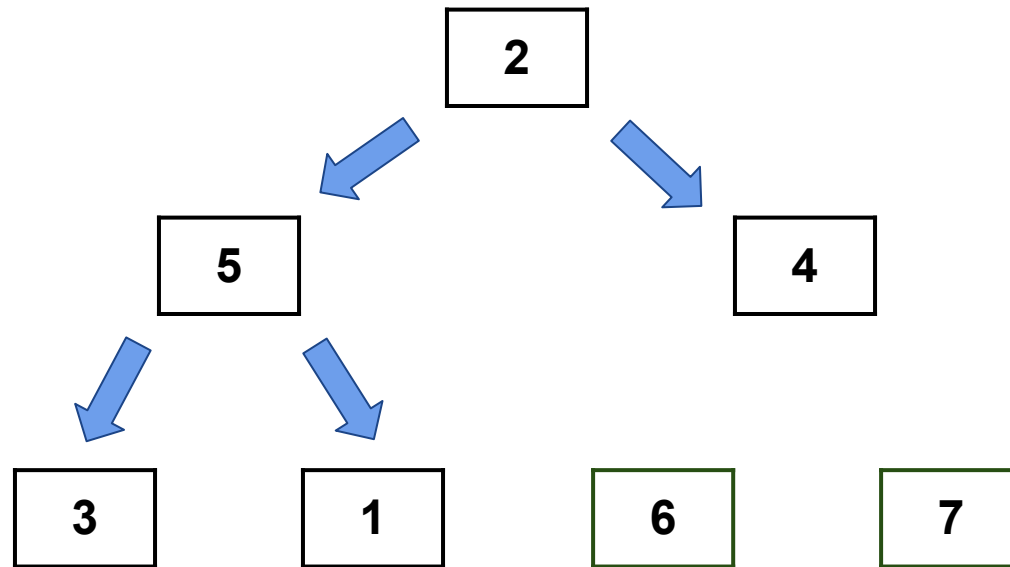


**Coloca a raiz no final
do sub-vetor**

Iteração 2

2	5	4	3	1	6	7
0	1	2	3	4	5	6

Árvore

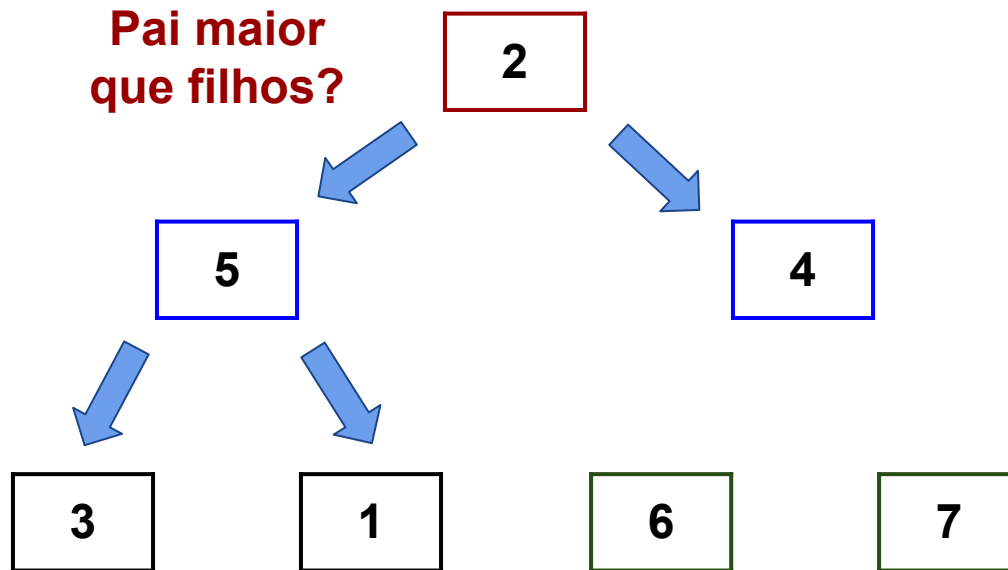


Já está na
posição certa

Iteração 2

2	5	4	3	1	6	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

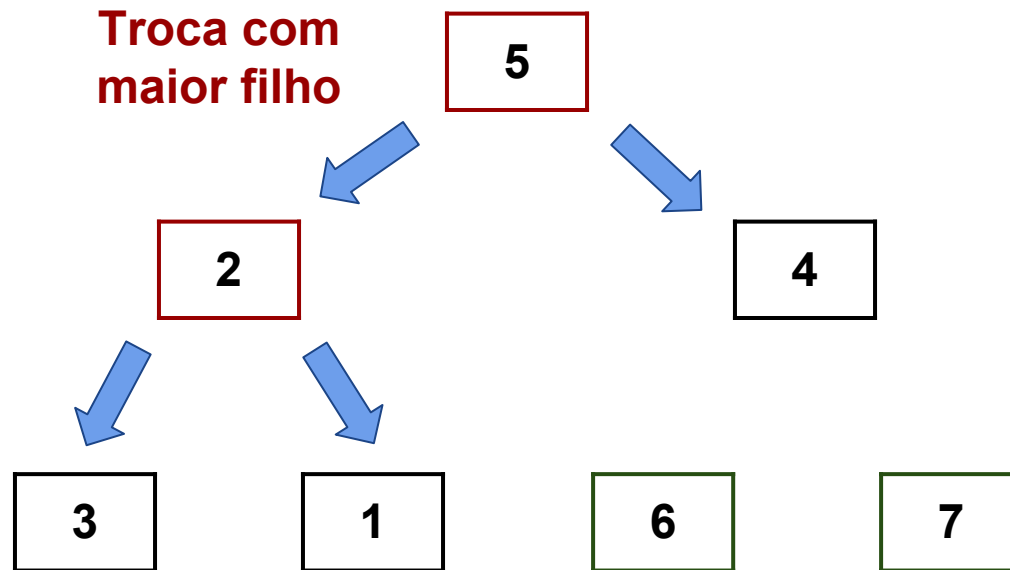
Árvore



Iteração 2

5	2	4	3	1	6	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

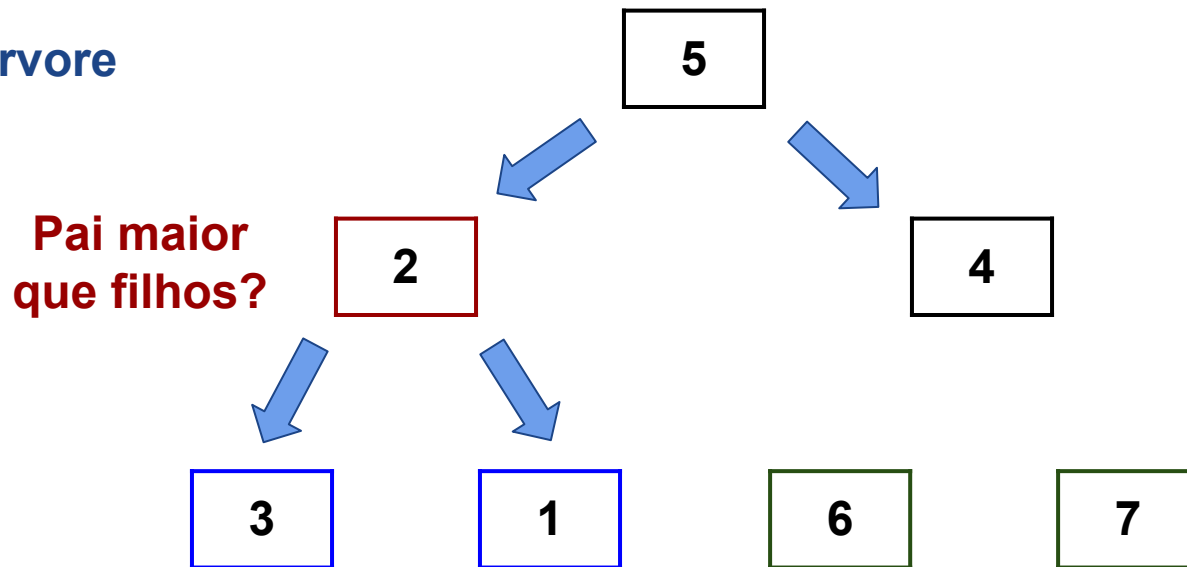
Árvore



Iteração 2

5	2	4	3	1	6	7
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

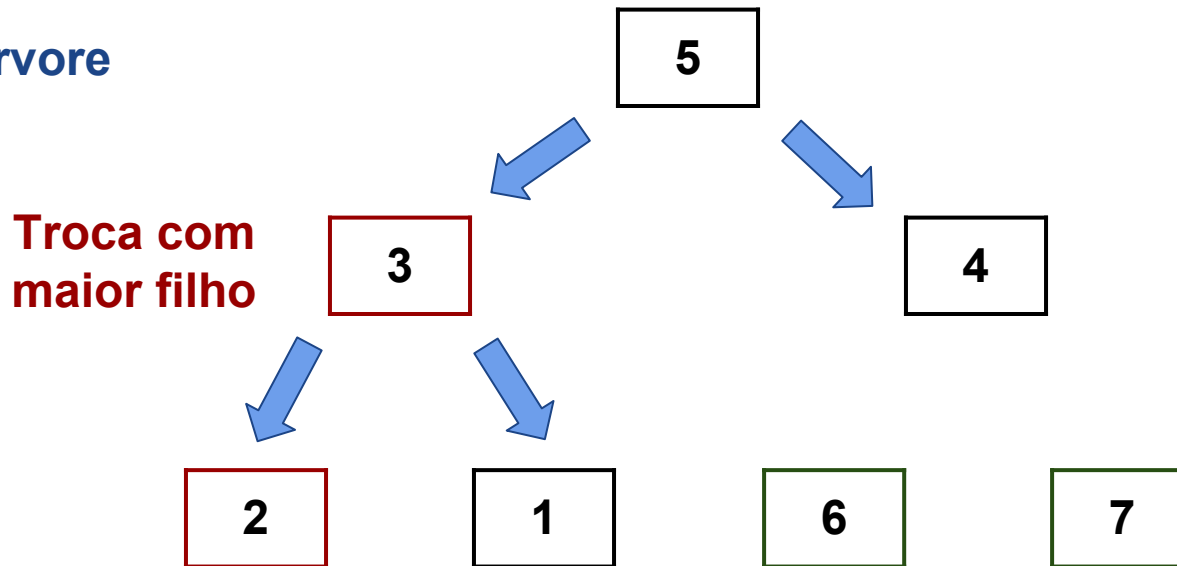
Árvore



Iteração 2

5	3	4	2	1	6	7
0	1	2	3	4	5	6
	i		$2i + 1$	$2i + 2$		

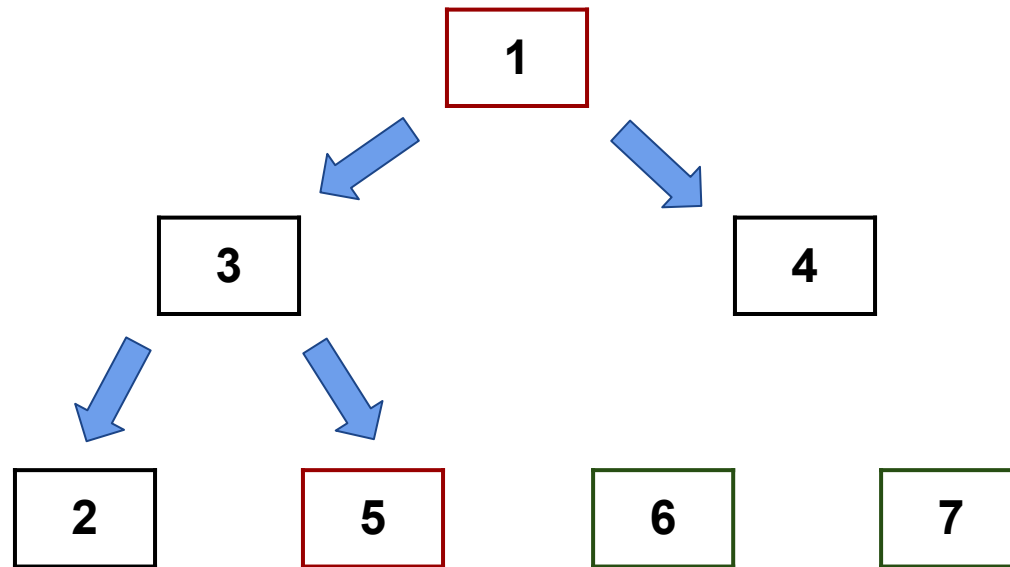
Árvore



Iteração 3

1	3	4	2	5	6	7
0	1	2	3	4	5	6

Árvore

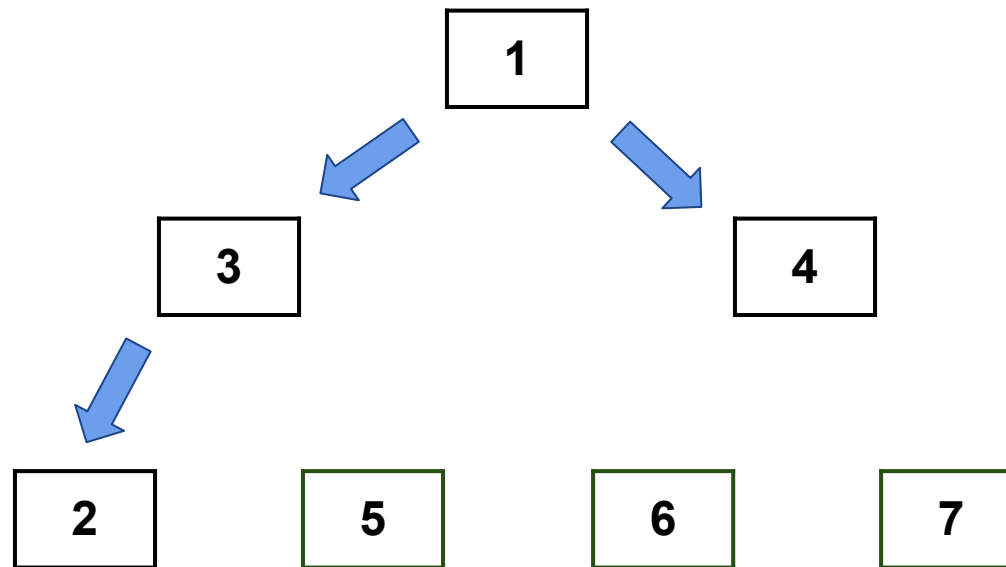


**Coloca a raiz no final
do sub-vetor**

Iteração 3

1	3	4	2	5	6	7
0	1	2	3	4	5	6

Árvore



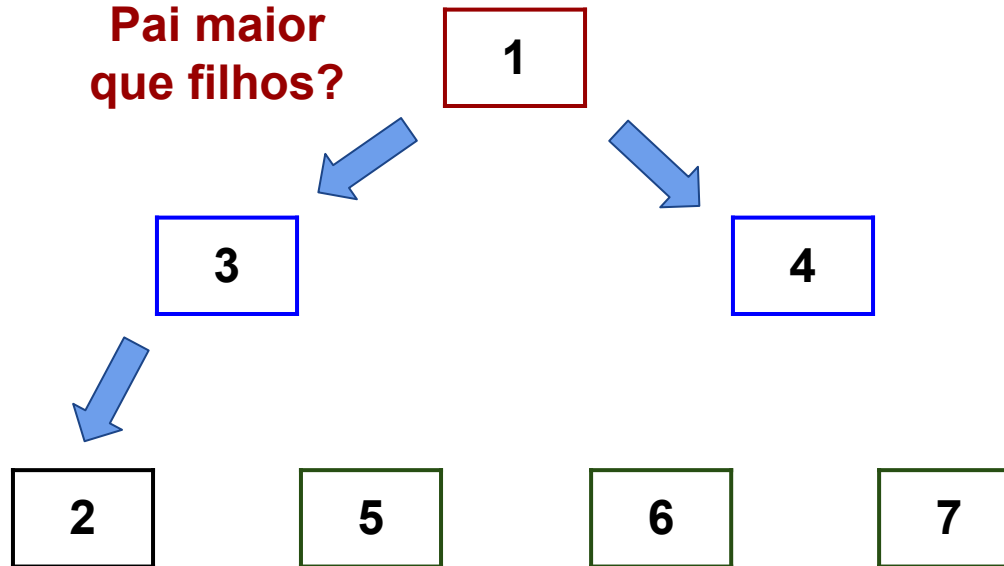
Já está na
posição certa

Iteração 3

1	3	4	2	5	6	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

Árvore

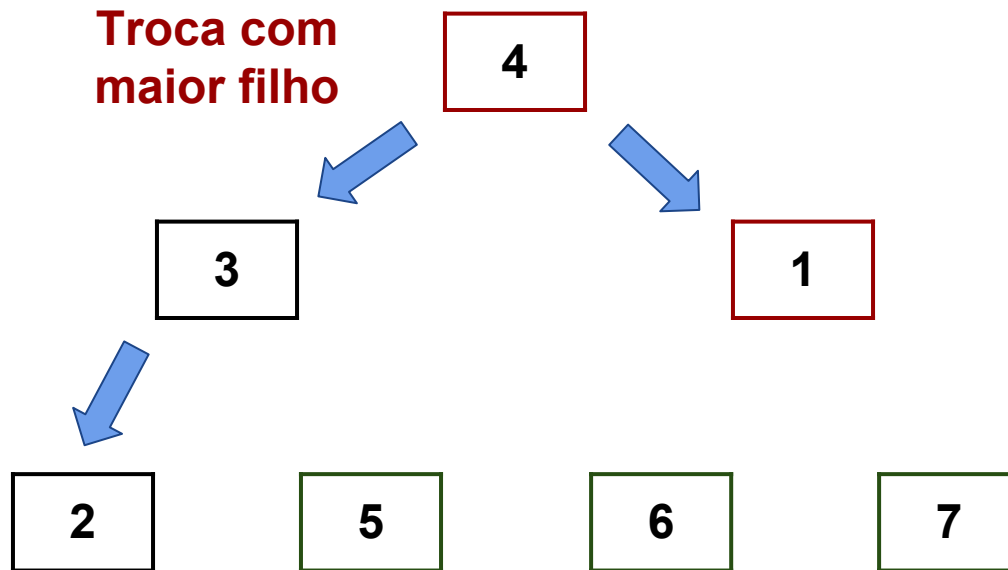
Pai maior
que filhos?



Iteração 3

4	3	1	2	5	6	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

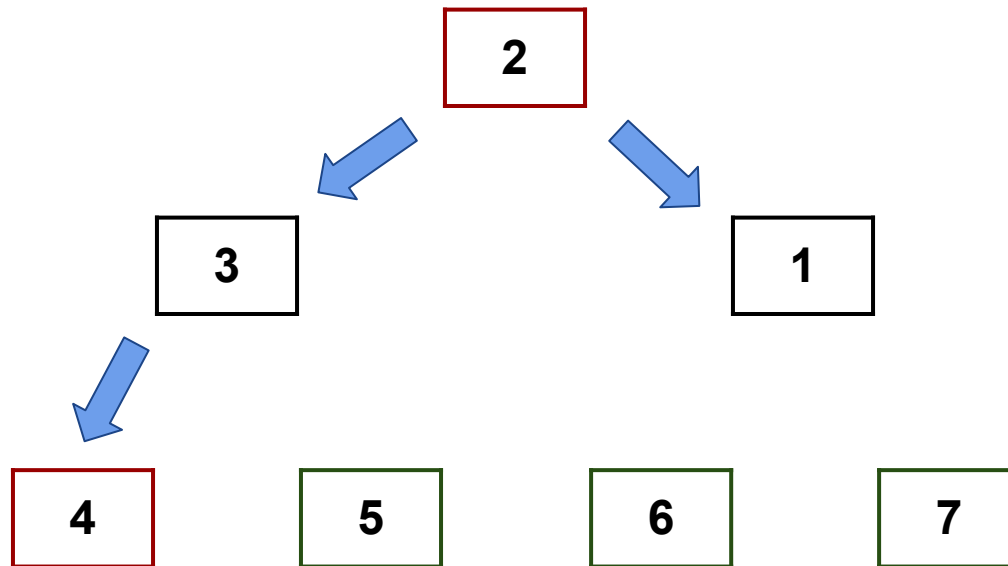
Árvore



Iteração 4

2	3	1	4	5	6	7
0	1	2	3	4	5	6

Árvore

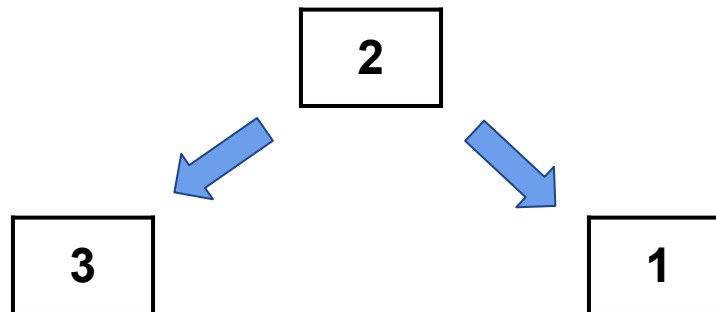


**Coloca a raiz no final
do sub-vetor**

Iteração 4

2	3	1	4	5	6	7
0	1	2	3	4	5	6

Árvore



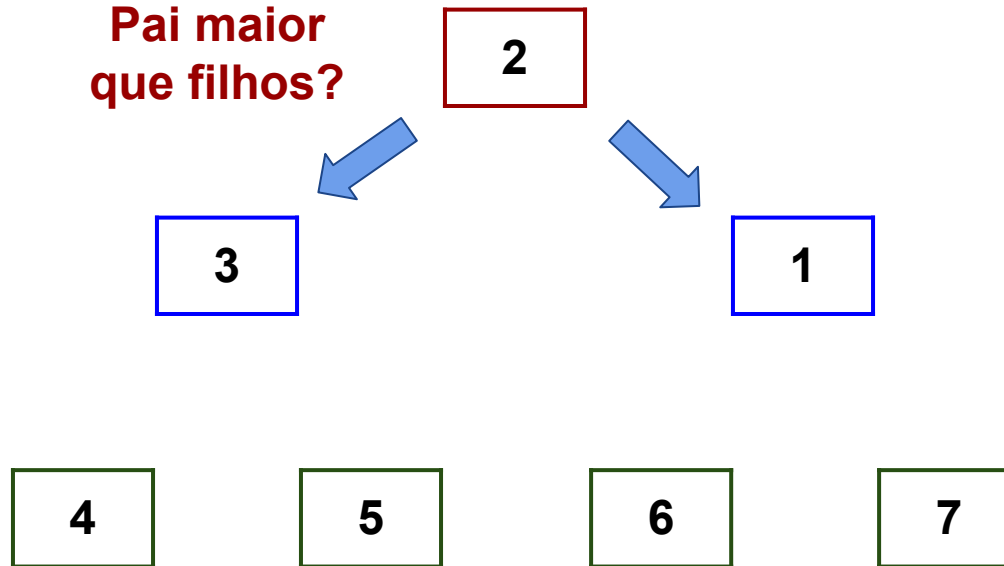
Já está na
posição certa

Iteração 4

2	3	1	4	5	6	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

Árvore

Pai maior
que filhos?

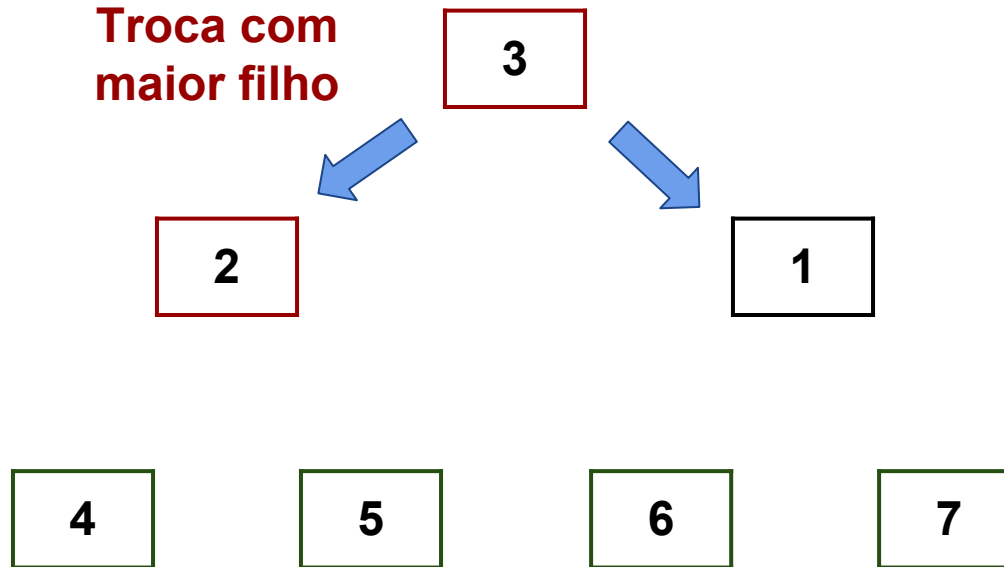


Iteração 4

3	2	1	4	5	6	7
0	1	2	3	4	5	6
i	$2i + 1$	$2i + 2$				

Árvore

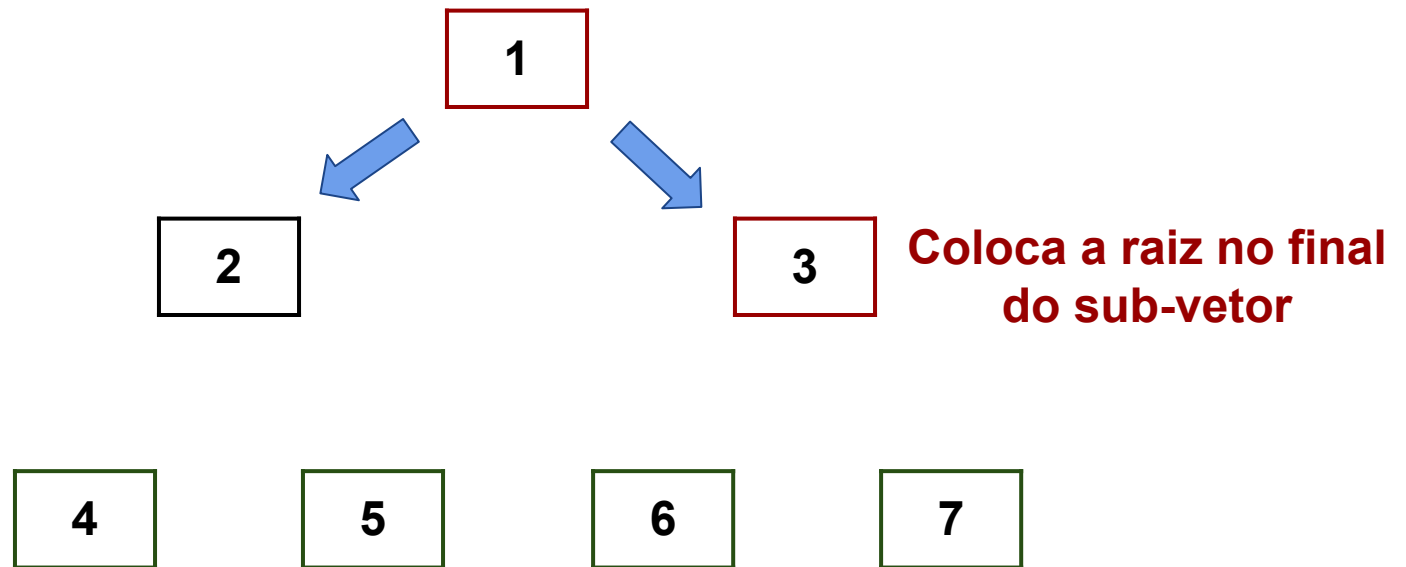
Troca com
maior filho



Iteração 5

1	2	3	4	5	6	7
0	1	2	3	4	5	6

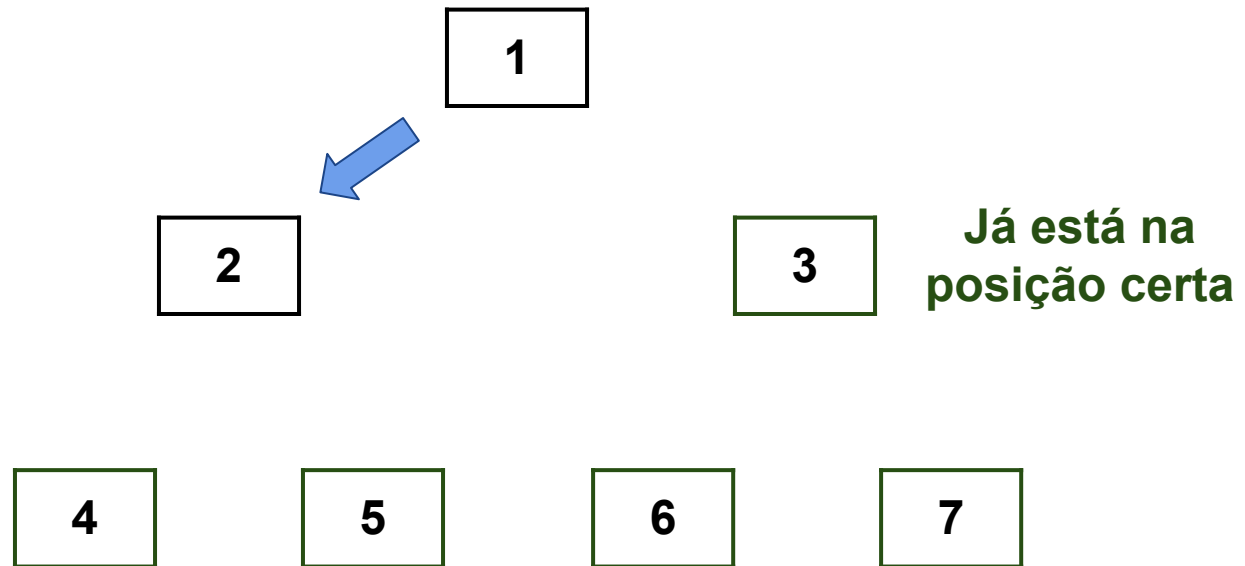
Árvore



Iteração 5

1	2	3	4	5	6	7
0	1	2	3	4	5	6

Árvore

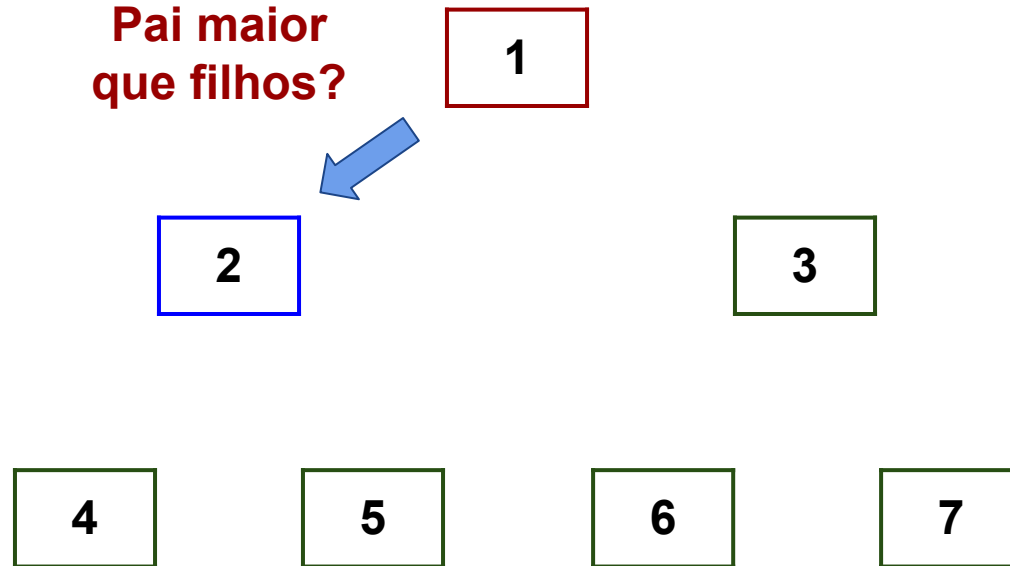


Iteração 5

1	2	3	4	5	6	7
0	1	2	3	4	5	6
i	$2i + 1$					

Árvore

Pai maior
que filhos?

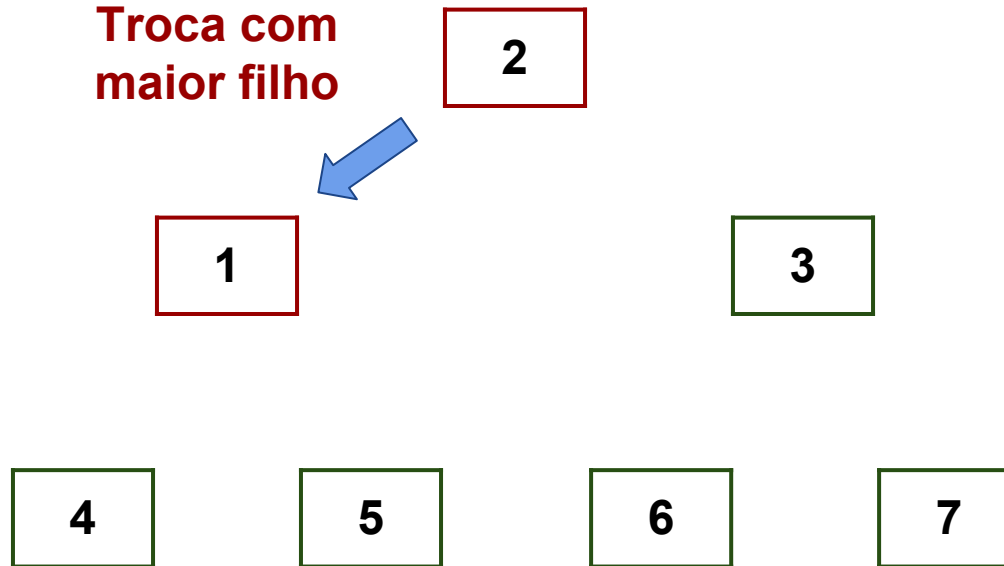


Iteração 5

2	1	3	4	5	6	7
0	1	2	3	4	5	6
i	$2i + 1$					

Árvore

Troca com
maior filho

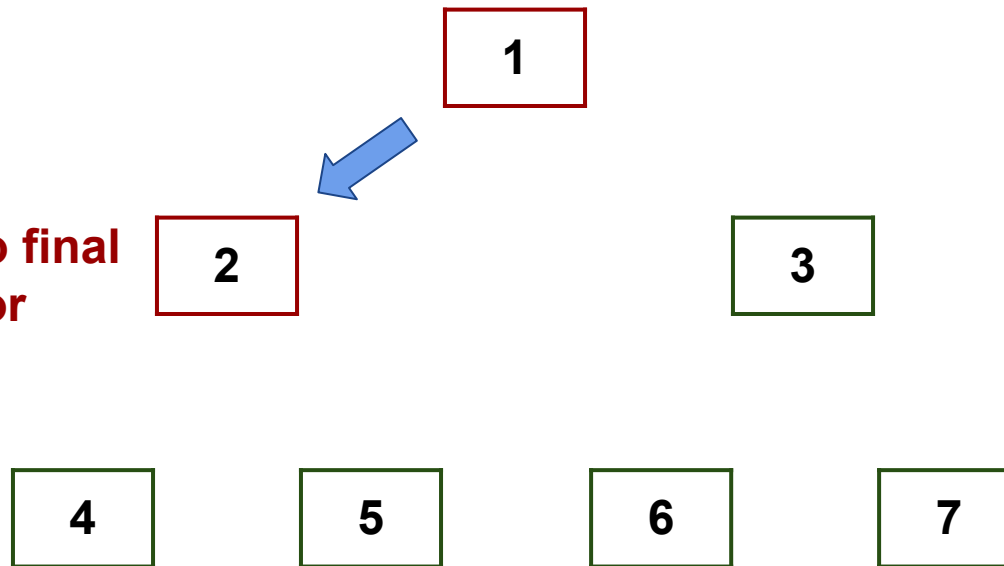


Iteração 6

1	2	3	4	5	6	7
0	1	2	3	4	5	6

Árvore

Coloca a raiz no final
do sub-vetor

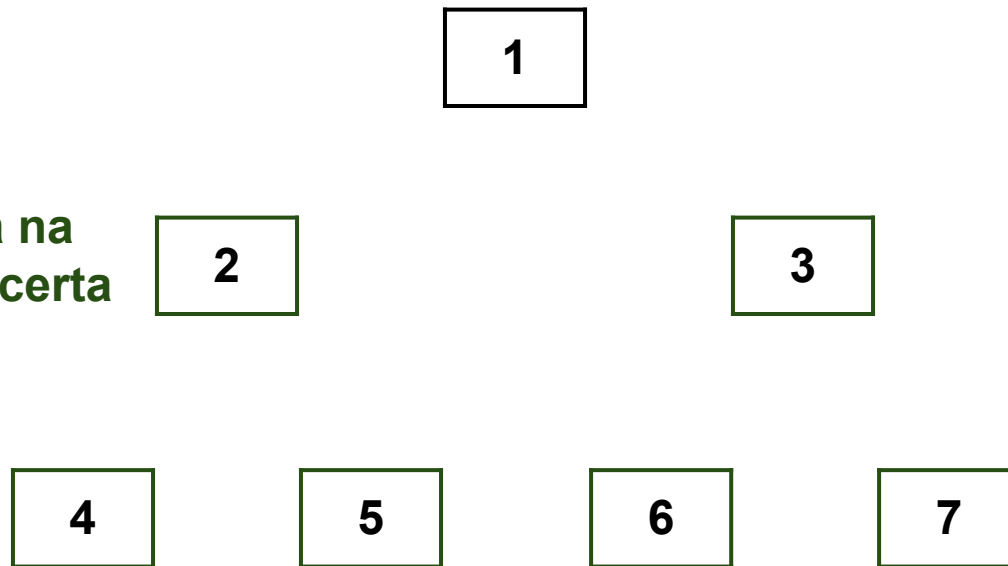


Iteração 6

1	2	3	4	5	6	7
0	1	2	3	4	5	6

Árvore

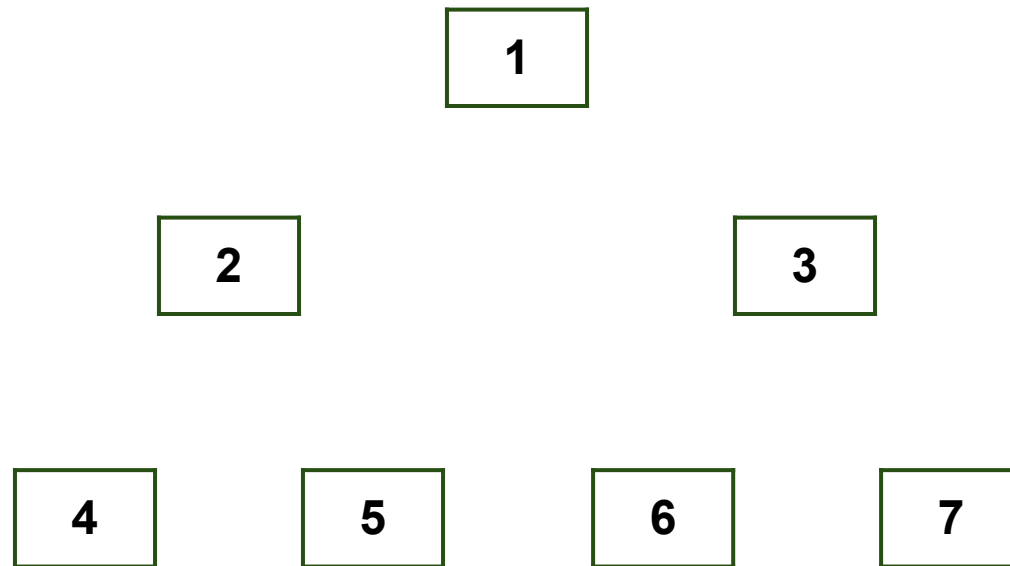
Já está na
posição certa



Resultado

1	2	3	4	5	6	7
0	1	2	3	4	5	6

Árvore



Implementação em sala



Características

- ▷ Faz parte da família de algoritmos por seleção;
- ▷ Bom desempenho;
- ▷ In place
- ▷ Não é Stable;
- ▷ Muito usado em sistemas operacionais;