

# Estruturas de Dados

Filas de Prioridades

# Filas de Prioridades

- O heapsort é um algoritmo excelente, mas uma boa implementação de quicksort normalmente o supera na prática.
- Não obstante, a estrutura de dados de heap pode ser empregada como uma fila de prioridades eficiente.
- Como ocorre no caso dos heaps, existem dois tipos de filas de prioridades: **máxima e mínima.**

# Filas de Prioridades

Uma fila de prioridades é uma estrutura de dados para manutenção de um conjunto  $S$  de elementos, cada qual com um valor associado chamado **chave**. Uma fila de prioridade **máxima** admite as operações a seguir:

- $\text{INSERT}(S, x)$
- $\text{MAXIMUM}(S)$
- $\text{EXTRACT-MAX}(S)$  remove e retorna o elemento de  $S$  com a maior chave
- $\text{INCREASE-KEY}(S, x, k)$  aumenta o valor da chave do elemento  $x$  para o novo valor  $k$ .

# Filas de Prioridades

HEAP-MAXIMUM(A)

**return** A[1]

HEAP-EXTRACT-MAX(A)

**if** tamanho-do-heap[A] < 1

**then erro** 'heap underflow'

max  $\leftarrow$  A[1]

A[1]  $\leftarrow$  A[tamanho-do-heap[A]]

tamanho-do-heap[A]  $\leftarrow$  tamanho-do-heap[A] - 1

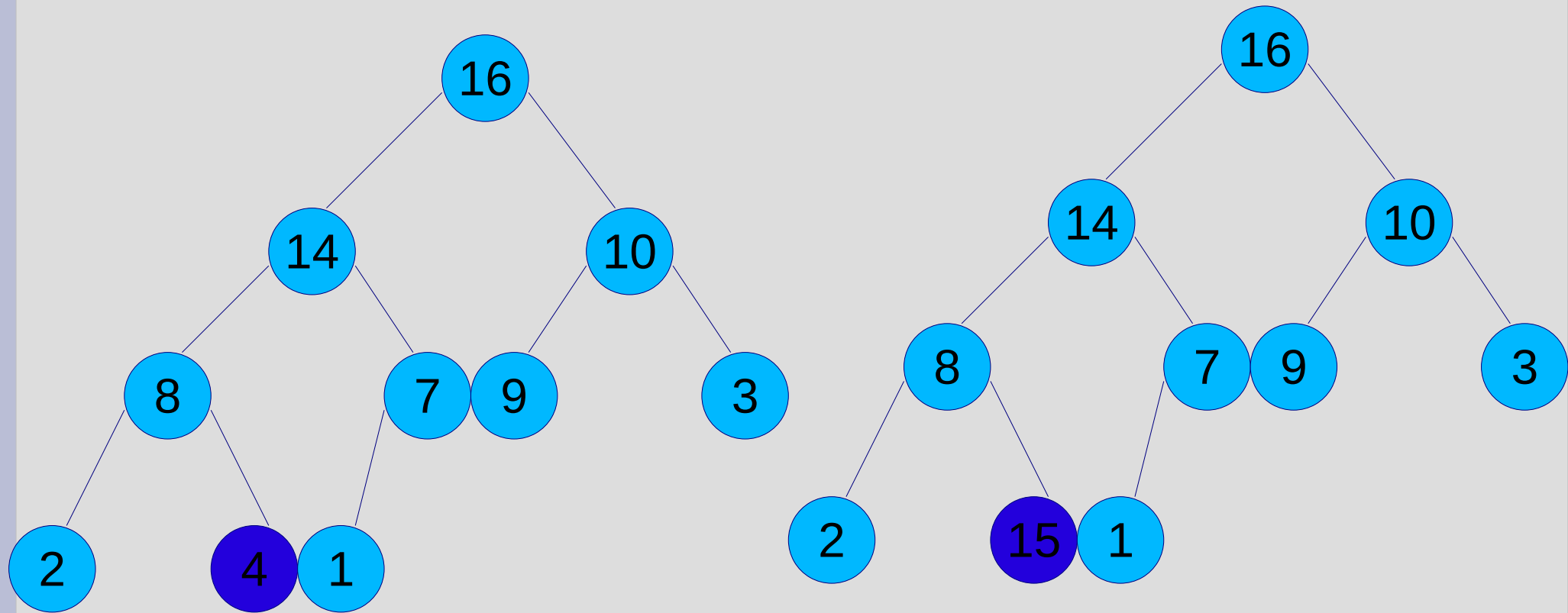
MAX-HEAPIFY(A, 1)

**return** max

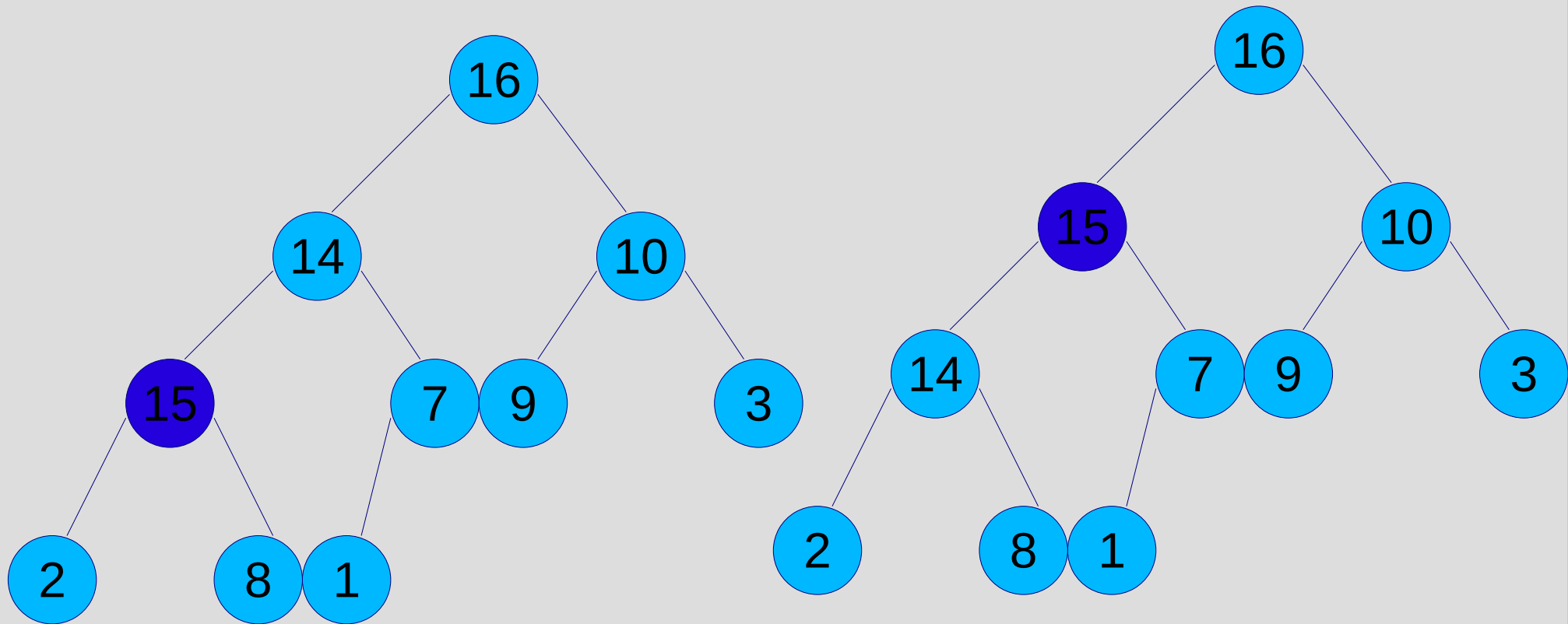
# Filas de Prioridades

```
HEAP-INCREASE-KEY(A,i,chave)
  if chave < A[i]
    then erro 'nova chave é menor que chave atual'
  A[i] ← chave
  while i>1 e A[PARENT(i)]<A[i]
    troca A[i] ↔ A[PARENT(i)]
    i ← PARENT(i)
```

# HEAP-INCREASE-KEY



# HEAP-INCREASE-KEY



# HEAP-INCREASE-KEY

O procedimento HEAP-INCREASE-KEY implementa a operação INCREASE-KEY. O elemento da fila de prioridades cuja chave deve ser aumentada é identificado por um índice  $i$  no arranjo.

Primeiro, o procedimento atualiza a chave do elemento  $A[i]$  para seu novo valor. Em seguida como o aumento da chave de  $A[i]$  pode violar a propriedade de heap máximo, o procedimento percorre um caminho desde esse nó em direção à raiz, até encontrar um lugar apropriado.



# MAX-HEAP-INSERT

```
MAX-HEAP-INSERT(A, chave)
    tamanho-do-heap[A]  $\leftarrow$  tamanho-do-heap[A] + 1
    A[tamanho-do-heap[A]]  $\leftarrow -\infty$ 
    HEAP-INCREASE-KEY(A, tamanho-do-heap[A], chave)
```

# MAX-HEAP-INSERT

O procedimento MAX-HEAP-INSERT implementa a operação INSERT. Ele toma como uma entrada a chave do novo elemento a ser inserido no heap máximo A. Primeiro, o procedimento expande o heap máximo, adicionando à árvore uma nova folha cuja chave é  $-\infty$ . Em seguida, ele chama HEAP-INCREASE-KEY para definir a chave desse novo nó com seu valor correto e manter a propriedade do heap máximo.

# MAX-HEAP-INSERT

O tempo de execução de MAX-HEAP-INSERT sobre um heap de  $n$  elementos é  $O(\lg n)$ .

Em resumo, um heap pode admitir qualquer operação de fila de prioridades em um conjunto de tamanho  $n$  no tempo  $O(\lg n)$ .

Slides baseados no livro **Algoritmos, Teoria e Prática**, de Cormen, T.H; Leiserson, C. E.; Rivest, R. L. e Stein, C.