

ENGENHARIA INFORMÁTICA E DE COMPUTADORES

Algoritmos e Estruturas de Dados

(parte 7 – Filas Prioritárias)

2º Semestre 2022/2023

Instituto Superior de Engenharia de Lisboa

Paula Graça

FILAS PRIORITÁRIAS (PRIORITY QUEUES)

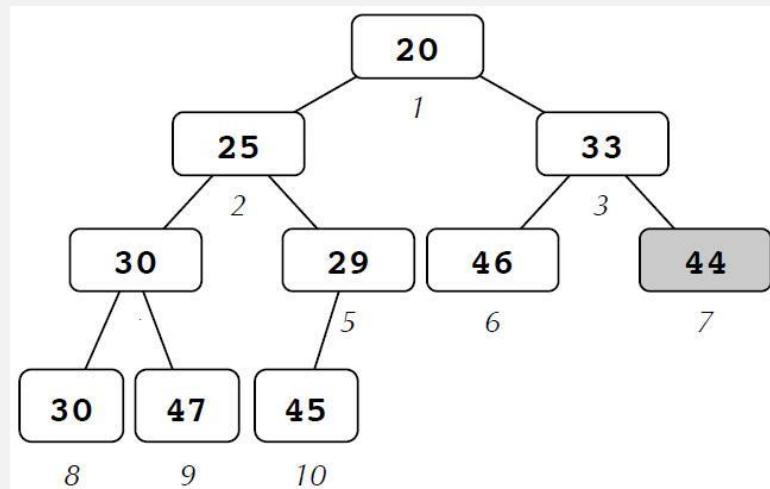
- Uma das aplicações mais populares dos *heaps* são as *filas prioritárias*



- Cada elemento é associado a um valor designado por chave (prioridade)
- O elemento com a chave de mais alta prioridade (maior ou menor valor) é o primeiro a ser extraído da fila
- *min-priority-queues* baseadas em *min-heaps*
- *max-priority-queues* baseadas em *max-heaps*

FILAS PRIORITÁRIAS – MIN-PRIORITY-QUEUE

- Uma fila prioritária (*min-priority-queue*), é uma estrutura de dados que mantém um conjunto Q de elementos, cada um com uma chave
- Suporta as seguintes operações:
 - **Peek**: devolve o elemento de Q mais prioritário (menor chave)
 - **Poll**: remove e devolve o elemento de Q mais prioritário (menor chave)
 - **Offer**: insere o elemento x no conjunto Q



FILAS PRIORITÁRIAS – MAX-PRIORITY-QUEUE

- Uma fila prioritária (*max-priority-queue*), é uma estrutura de dados que mantém um conjunto Q de elementos, cada um com uma chave
- Suporta as seguintes operações:
 - **Peek(Q)**: devolve o elemento de Q mais prioritário (maior chave)
 - **Poll(Q)**: remove e devolve o elemento de Q mais prioritário (maior chave)
 - **Offer(Q, x)**: insere o elemento x no conjunto Q

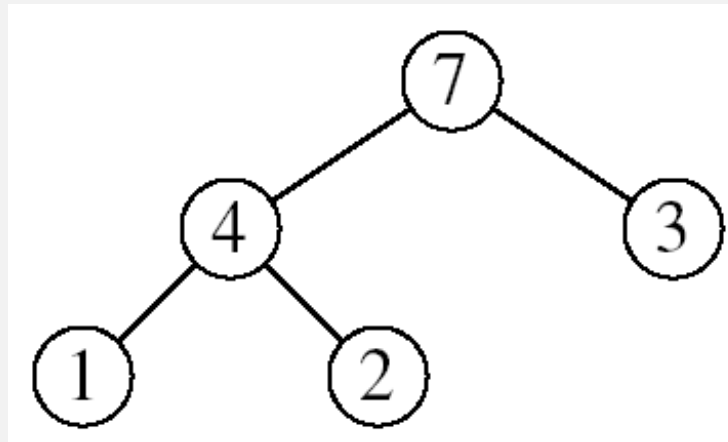
PEEK

- Considerando uma fila prioritária implementada num **Max-heap**
- **Peek** - devolve o valor do *heap* mais prioritário (maior chave)

```
Peek (H)  
return H[1]
```

Tempo de execução: **$O(1)$**

Heap H:

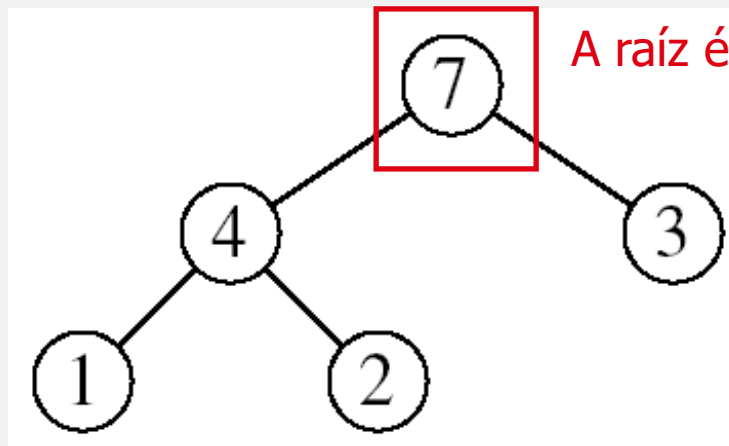


Peek(H) returns 7

POLL

- **Poll** - Extraí removendo o elemento do *heap* mais prioritário (maior chave)
 - Extraí o elemento da raiz e substitui-o pelo último
 - Decrementa a dimensão do *max-heap*
 - Executa Max-Heapify na raiz (*max-heap* de dimensão $n - 1$)

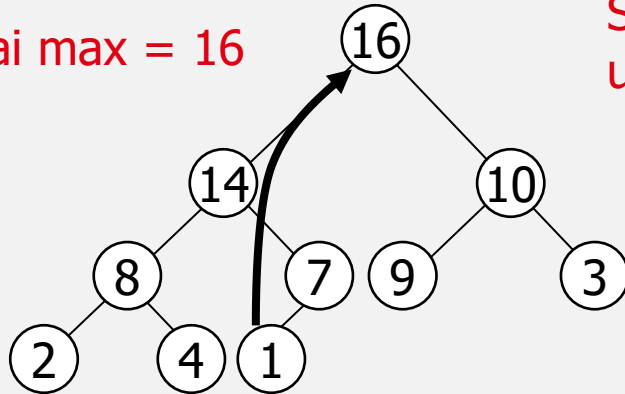
Heap H:



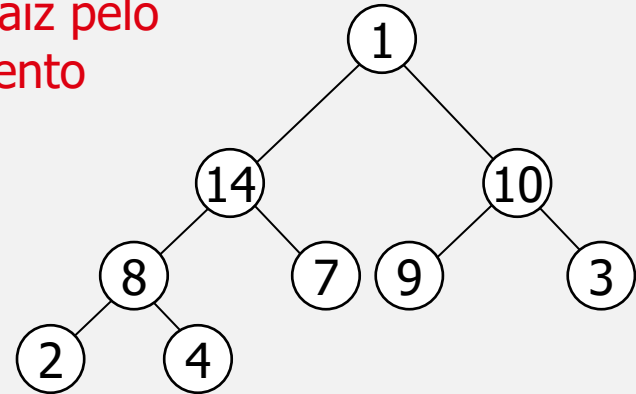
A raiz é o maior elemento

POLL - EXEMPLO

Extrai max = 16

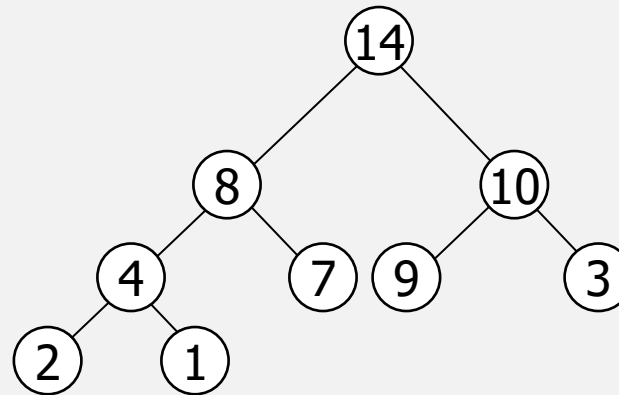


Substitui a raiz pelo ultimo elemento



Decrementa a dimensão do heap

Call Max-Heapify (H, 1, n-1)



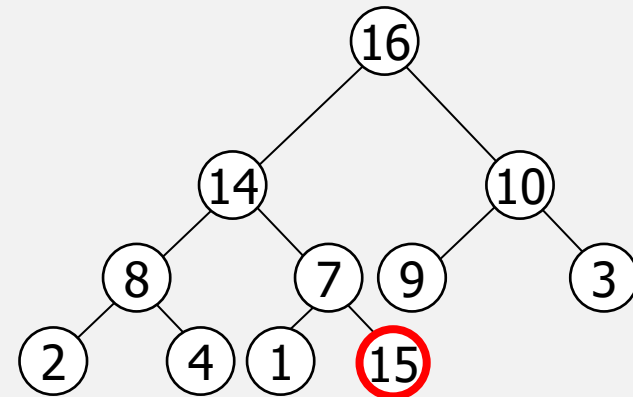
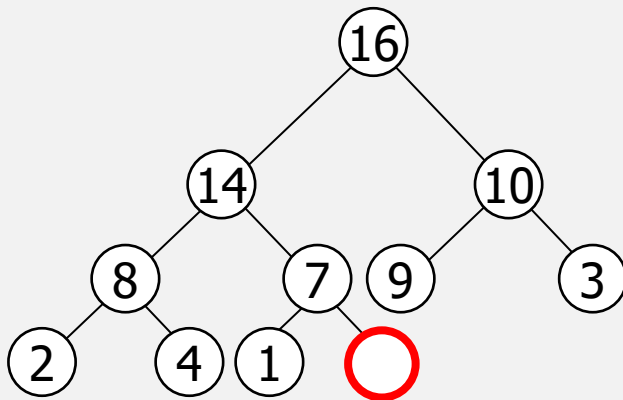
POLL - ALGORITMO

```
Poll (H)
  if H.heap-size < 1
    error "heap underflow"
  max = H[1]
  H[1] = H[H.heap-size]
  H.heap-size = H.heap-size - 1
  Max-Heapify(H, 1)
  return max
```

Tempo de execução: $O(\lg n)$

OFFER

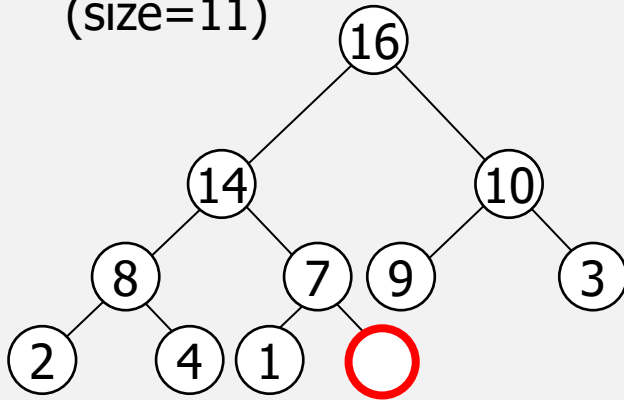
- **Offer** - insere um novo elemento no *heap*
 - Expande o *max-heap* com um novo elemento
 - Atualiza o elemento com a nova chave
 - Executa Heap-Increase-Key para manter a propriedade de ordenação do *max-heap*



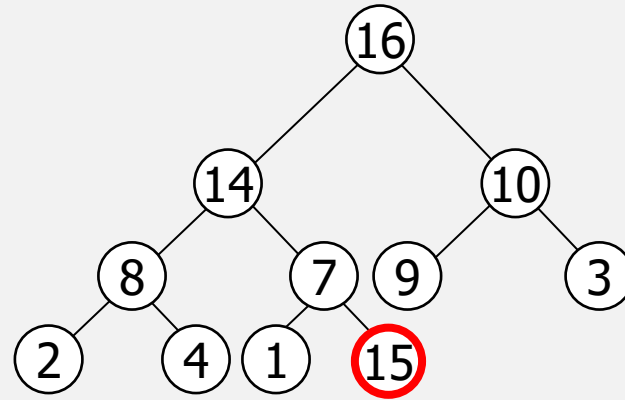
Key [size-heap+1] \leftarrow 15

OFFER - EXEMPLO

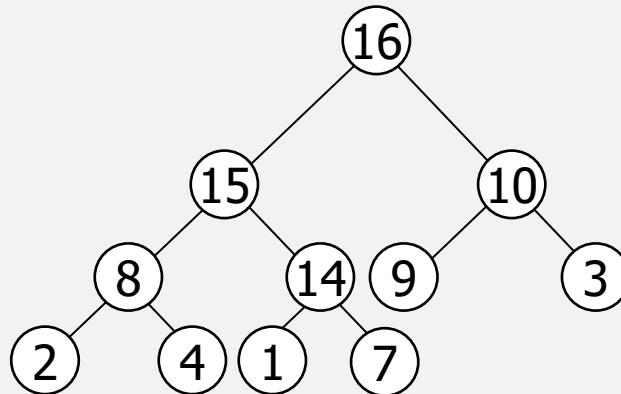
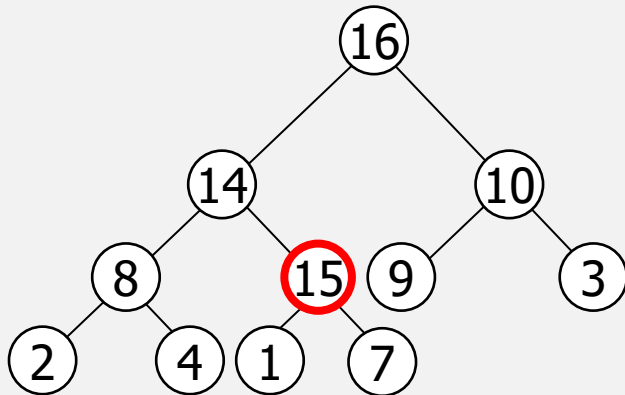
Expande o *heap* com um novo nó incrementando a sua dimensão (size=11)



Atualiza a chave do novo nó para 15
Executa Heap-Increase-Key em H[11]



O *heap* contém o novo elemento adicionado



OFFER - ALGORITMO

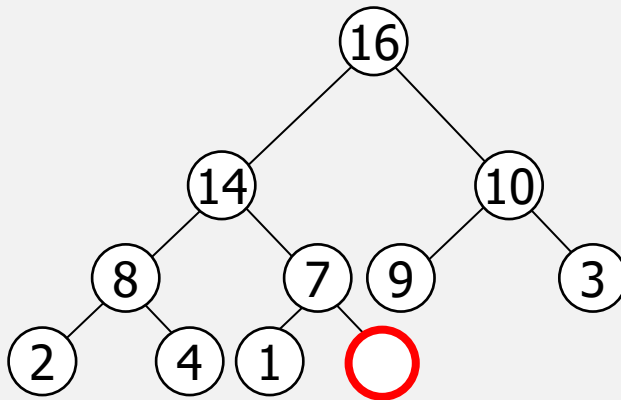
Offer (H, key)

$H.\text{heap-size} = H.\text{heap-size} + 1$

$H[H.\text{heap-size}] = \text{key}$

Heap-Increase-Key(H, H.heap-size)

Tempo de execução: $O(\lg n)$



JAVA PRIORITY QUEUE

- ○ Java suporta a estrutura *PriorityQueue* implementada num min-heap
- Principais operações:

Insert Elements to PriorityQueue

- `add()` - Inserts the specified element to the queue. If the queue is full, it throws an exception.
- `offer()` - Inserts the specified element to the queue. If the queue is full, it returns `false`.

Access PriorityQueue Elements

To access elements from a priority queue, we can use the `peek()` method. This method returns the head of the queue.

Remove PriorityQueue Elements

- `remove()` - removes the specified element from the queue
- `poll()` - returns and removes the head of the queue

JAVA PRIORITY QUEUE

Other PriorityQueue Methods

Methods	Descriptions
<code>contains(element)</code>	Searches the priority queue for the specified element. If the element is found, it returns <code>true</code> , if not it returns <code>false</code> .
<code>size()</code>	Returns the length of the priority queue.
<code>toArray()</code>	Converts a priority queue to an array and returns it.