



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA



DISIM
Dipartimento di Ingegneria
e Scienze dell'Informazione
e Matematica



Laboratorio di Algoritmi e Strutture Dati a.a. 2022/2023

Heap binari e la classe PriorityQueue

Giovanna Melideo
Università degli Studi dell'Aquila
DISIM

Coda di priorità

- È una raccolta in cui può essere rimosso soltanto l'elemento con **priorità massima**, dato un criterio di confronto fra gli elementi
- `Java.util` contiene la classe `PriorityQueue<E>`
- Gli **heap binari (Min-Heap)** costituiscono l'elemento centrale per la realizzazione della classe `PriorityQueue<E>`.

PriorityQueue<E>: constructors

- **PriorityQueue()** Creates a PriorityQueue with the default initial capacity (11) that orders its elements according to their natural ordering.
- **PriorityQueue(Collection<? extends E> c)** Creates a PriorityQueue containing the elements in the specified collection.
- **PriorityQueue(Comparator<? super E> comparator)** Creates a PriorityQueue with the default initial capacity and whose elements are ordered according to the specified comparator.
- **PriorityQueue(int initialCapacity)** Creates a PriorityQueue with the specified initial capacity that orders its elements according to their natural ordering.
- **PriorityQueue(int initialCapacity, Comparator<? super E> comparator)** Creates a PriorityQueue with the specified initial capacity that orders its elements according to the specified comparator.
- **PriorityQueue(PriorityQueue<? extends E> c)** Creates a PriorityQueue containing the elements in the specified priority queue.
- **PriorityQueue(SortedSet<? extends E> c)** Creates a PriorityQueue containing the elements in the specified sorted set.

PriorityQueue<E>: methods

- **boolean add(E e)** Inserts the specified element into this priority queue.
- **void clear()** Removes all of the elements
- **Comparator<? super E> comparator()** Returns the comparator used to order the elements in this queue, or null if this queue is sorted according to the natural ordering of its elements.
- **boolean contains(Object o)** Returns true if this queue contains the specified element.
- **Iterator<E> iterator()** Returns an iterator over the elements in this queue.
- **boolean offer(E e)** Inserts the specified element into this priority queue.
- **E peek()** Retrieves, but does not remove, the head of this queue (returns null if the queue is empty)
- **E poll()** Retrieves and removes the head of this queue, or returns null if this queue is empty.
- **boolean remove(Object o)** Removes a single instance of the specified element from this queue, if it is present (Remark: viola la definizione di priority queue!)
- **int size()** Returns the number of elements in this collection.
- **Object[] toArray()** Returns an array containing all of the elements in this queue.

PriorityQueue<E>: esempi di uso

- Esempi di uso della classe `PriorityQueue<E>` (JCF)
 - Focus sulle possibili violazioni nell'ordine di accesso agli elementi
 - `StringPriorityQueue.java`
 - Implementazione alternativa e meno efficiente del metodo di ordinamento Heapsort
 - `HeapSort_ByPriorityQueue.java`

La classe `MyPriorityQueue<E>`

- Semplice implementazione didattica dell'interfaccia `MyQueue<E>` basata su heap binari:
 - `MyPriorityQueue.java`
- Esempio d'uso della classe
 - `Test_MyPriorityQueue.java`



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA



DISIM
Dipartimento di Ingegneria
e Scienze dell'Informazione
e Matematica



Domande?

Giovanna Melideo
Università degli Studi dell'Aquila
DISIM