# Relazione su Priority Queue

$$Autore : Georgiev\ David\ mat : 1043306$$

## Introduzione

In questa relazione viene esaminata la struttura della coda di priorità generica implementata attraverso uno heap minimo e con l'ausilio di altre strutture dati implementati in librerie native a Java. In particolare modo, nella implementazione concreta fornita dal sottoscritto, si è fatto uso di un ArrayList per rappresentare lo heap minimo ovvero è un array che ha come elemento in prima posizione quello che precede tutti gli altri elementi. Tale ordine è determinato dal comportamento del comparatore passato in input al costruttore della coda stessa.

La coda mantiene altre due invarianti sempre vere durante la sua esistenza;

$$\forall i \in \{0, 1, \ldots, n-1\} \quad \text{se } 0 < 2i+1 < n, \text{ allora } P(A[i]) > P(A[2i+1])$$

$$\forall i \in \{0, 1, \ldots, n-1\} \quad \text{se } 0 < 2i+2 < n, \text{ allora } P(A[i]) > P(A[2i+2])$$

Ovvero il padre ha sempre priorità maggiore rispetto ai figli.

Per migliorare la complessità nel caso medio di questa coda, inoltre, si è utilizzata una HashMap che utilizza come chiavi gli elementi della coda e come valori le loro rispettive posizioni: in questo modo si ha sempre un diretto accesso alla posizione di ogni elemento consentendo ad alcuni metodi un miglioramento in termini di complessità temporale. Il prezzo di questa agevolazione consiste nel memorizzare sia i hashcode dei valori nella coda che un numero intero corrispondente alla loro posizione.

## Osservazioni

Questa coda di priorità non può contenere due elementi di priorità uguale quindi ci potrebbero essere incongruenze tra il comparator e il metodo equals se implementati in modo che si contrastino. Siccome viene usata anche una

HashMap sarebbe opportuno che il metodo hashCode e il metodo equals siano definiti in modo coerente e che non si contrastino. Da tutto queste affermazioni si deduce che i tre metodi equals, comparator e hashCode siano legati tra di loro ed è indispensabile che la loro implementazioni siano corrette.