Coda con Priorità

Caratteristiche	Le **code con priorità** sono un caso particolare di **insieme** dove, sugli elementi, è definita una **relazione di ordinamento totale** ≤. In questo tipo di struttura le **operazioni consentite** sono: - inserire un nuovo elemento (in coda); - estrarre l'elemento minimo (dalla testa). Gli elementi dovranno essere ordinati (e serviti) secondo priorità: questa sarà una proprietà associata agli elementi.
Rappresentazioni	Una cosa con priorità può essere realizzata usando: - **Lista Ordinata** - **Lista non Ordinata** - **Albero Binario** Rappresentazione più usata, dal momento che, seppur gli elementi sono in relazione tra loro, non c'è una relazione strutturale sull'insieme delle posizioni. - **Heap**
Rappresentazione con Albero Binario	Nella rappresentazione con albero binario, questo deve avere alcune proprietà: - **Quasi Perfettamente Bilanciato** Se ***k*** è il livello massimo delle foglie, allora l'albero ha esattamente $2^k - 1$ nodi di livello minore di ***k***. In altre parole la differenza tra sottoalbero sinistro e destro deve essere al più di una foglia. - **Parzialmente Ordinato** Ogni nodo contiene un elemento che è maggiore di quello del padre. Le operazioni di inserimento ed eliminazione devono tenere conto di queste due proprietà che devono essere sempre rispettate; infatti dovranno essere eseguite in due steps: 1. inserimento/cancellazione del nodo; 2. modifica dell'albero in modo tale che le proprietà vengano rispettate.
Rappresentazione con Heap	L'**heap** è una struttura dati basata sugli alberi dove, se A è un genitore di B, allora la chiave (cioè il valore) di A è ordinata in base al valore di B, in base alla relazione d'ordine applicata all'intero heap. Da questa definizione abbiamo **maxHeap**, dove le chiavi di un nodo sono sempre maggiori o uguali a quelle dei figli e **minHeap**, dove vale il contrario. L'heap può anche essere descritto tramite **array**; avremo che gli indici assunti per i vari nodi saranno: - **padre** = $i/2$ - **nodo sinistro** = $2i$ - **nodo destro** = $2i + 1$

	Gli heap sono generalmente implementati tramite array (di dimensione fissa, o variabile) e non richiede puntatori fra gli elementi. Dopo la rimozione, inserimento o sostituzione di un elemento, la proprietà di heap potrebbe essere violata, rendendo necessario il bilanciamento tramite operazioni interne. Gli **heap binari** (heap sviluppati su alberi binari) possono essere rappresentati in un modo molto efficiente (dal punto di vista dello spazio) utilizzando un solo array. Il primo elemento rappresenta la radice. I due elementi seguenti contengono i figli della radice. I quattro seguenti contengono i figli dei figli, e così via. In particolare, per la realizzazione con heap della coda con priorità, si utilizzano proprio gli heap binari.

Specifica Coda di Priorità

TIPI:

- PrioriCoda: insieme di code con priorità con elementi di tipo tipoelem

OPERATORI:

creaPrioriCoda() = A

POST: $A = \emptyset$

inserisci(x,A) = A'

POST: $A' = A \cup \{x\}$ (se $x \in A$ allora $A = A'$)

min(A) = x

PRE: $A \neq \emptyset$

POST: $x \in A$ and $x < y$ per ogni $y \in A$ $(x \neq y)$

cancellaMin(A) = A'

PRE: $A \neq \emptyset$

POST: $A' = A \setminus \{x\}$, $x = \min(A)$