

Esercitazione 5

Programmazione Avanzata ed Elementi di Ingegneria del Software

Esercizio 1:

Implementare, in modo sia ricorsivo che procedurale, le funzioni per gestire un heap.

Sono necessarie le funzioni:

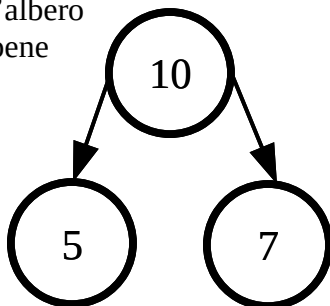
1. la funzione `max_heapify(array, size, i)`, che posiziona l'elemento in posizione *i*-esima nella posizione corretta all'interno del vettore affinché quest'ultimo non violi la proprietà "max_heap".
2. la funzione `build_max_heap(array, size)`, che ordina il vettore in modo che esso rappresenti un albero binario che rispetti la proprietà "max_heap".
3. La funzione `heap_sort(array, size)`.

Esercizio 2:

Implementare una qualunque di queste funzioni usando però un approccio "min-heap".

*La proprietà di max_heap in un albero binario afferma che un nodo dentro l'albero deve avere un valore superiore (o uguale) dei suoi nodi figli. Ad esempio:

quest'albero
va bene



il nodo "7" viola
la max_heap

