

Laboratorio di Algoritmi e Strutture Dati
2015/16
Progetto Totale, Codice: PT1

Docente: Luca Tesei

Per la scadenza si vedano le date definite per ogni appello nel wiki del corso

Testo progetto

N.B.: Questo progetto può essere sviluppato singolarmente o in un gruppo di massimo 2 studenti.

- Si definisca una classe `AVLTree<E extends Comparable<E>>` che implementi tutti i metodi che si ritengono necessari per:
 - costruire (vuoto o da un insieme di nodi),
 - inserire un nodo in,
 - estrarre (cancellare) un nodo da,
 - restituire il nodo massimo rispetto all'ordinamento di `E` da,
 - restituire il nodo minimo rispetto all'ordinamento di `E` da,
 - ricercare in

alberi AVL, cioè alberi binari di ricerca bilanciati il cui il coefficiente di bilanciamento $b(n)$ per ogni nodo n vale 1, 0 o -1. Il coefficiente di bilanciamento è definito come la differenza tra l'altezza del sottoalbero sinistro e quella del sottoalbero destro di un nodo:

$$b(n) = h(n.left) - h(n.right)$$

Nel caso di un nodo foglia il coefficiente di bilanciamento vale 0. Nel caso di sottoalbero sinistro (o destro) non esistente l'altezza $h(n.left)$ (o $h(n.right)$) da considerare per il calcolo del coefficiente è zero.

Le operazioni di inserimento e cancellazione dovranno mantenere la proprietà dell'albero AVL tramite le opportune operazioni di rotazione.

- Si definisca una classe di test `AVLTree<E extends Comparable<E>>` che mostri che tutte le operazioni sono state implementate correttamente.

- Si utilizzi la classe `AVLTree<E extends Comparable<E>>` sviluppata in precedenza per scrivere una classe che implementi tutti i metodi richiesti dall'interfaccia `java.util.SortedSet<E>`.
- Si definisca una classe di test per la classe implementata che mostri che tutte le operazioni richieste dall'interfaccia `java.util.SortedSet<E>` sono state implementate correttamente.
- Si scriva una relazione in cui si descrivono le strategie che si sono adottate per l'implementazione del codice richiesto.
- Si discuta nella relazione anche la realizzazione di un algoritmo di ordinamento basato su alberi AVL. Si cerchi di valutarne la complessità nel caso ottimo, medio e pessimo rispetto ad altri algoritmi di ordinamento basati su confronti. Le valutazioni possono anche essere fatte (ma non è obbligatorio) implementando l'algoritmo progettato e usando il framework di valutazione numerica delle prestazioni di algoritmi di ordinamento presentato a lezione (si faccia riferimento al wiki del corso per l'anno accademico 2015/16).

Le classi implementate dovranno essere completamente autodocumentate tramite commenti interpretabili dall'utility `javadoc` e con commenti privati. Codice non adeguatamente commentato sarà valutato negativamente.

Modalità di Consegna

I file `.java` per le classi, senza indicazione di package (cioè appartenenti al package di default), e il file con la relazione in pdf, devono essere caricati entro la data di scadenza in una cartella Google Drive dal nome

`ASDL1516PT1-APPX-CognomeStudente-NomeStudente`

dove X è il numero dell'appello in cui si consegna il progetto. La cartella deve essere condivisa, in sola lettura, tramite l'account

`nome-studente.cognome-studente@studenti.unicam.it`

di uno degli studenti del gruppo con gli account:

- `luca.tesei@unicam.it`
- `emanuela.merelli@unicam.it`

e con gli account degli altri studenti del gruppo.

Per la scadenza, farà fede la data dei file su Google Drive.