



Commento al Laboratorio n. 11

Esercizio n.1: Titoli azionari

ADT e quasi ADT per dati e collezioni di dati sono tutti realizzati su 2 moduli `.h` e `.c`.

Strutture dati:

- **Quasi ADT Data:** tipologia 1 con 3 campi interi per anno, mese e giorno con funzioni di lettura (`DATAload`), memorizzazione (`DATAstore`) e confronto (`DATAcmp`)
- **Quasi ADT Ora:** tipologia 1 con 2 campi interi per ora e minuti con funzioni di lettura (`ORAlload`) e memorizzazione (`ORAstore`)
- **ADT di I classe Titolo:** struct con campo di tipo `TitoloKey`, 2 campi di tipo `int` per quotazioni minima e massima e puntatore `bstQuotazioni` a un BST delle quotazioni per quel titolo. Le funzioni, oltre a creare un titolo nuovo (`TITOLOnew`) e a memorizzarlo (`TITOLOstore`), permettono di estrarne la chiave, cioè il codice del titolo (`TITOLOgetKey`), di confrontare i codici di 2 titoli (`TITOLOcmp`) e di inserire una nuova transazione (`TITOLOinsertTransazione`). Dato il BST relativo ad un titolo, la funzione `TITOLOgetQuotazione` permette di recuperare la quotazione in una certa data, le funzioni `TITOLOminAssoluto` e `TITOLOmaxAssoluto` ritornano la quotazione minima/massima per tutte le date, mentre la funzione `TITOLOminmaxRange` lo fa per un intervallo di date utilizzando la `BSTquotazioniMinmaxRange`. La funzione `TITOLOtreeBalance` permette di ribilanciare il BST di un certo titolo utilizzando la `BSTquotazioniBalance`. Tutte queste ultime funzioni si appoggiano sul BST della collezione di dati `Quotazioni`
- **ADT di I classe LISTtitoli** per la collezione di dati di tipo `Titolo` implementato come lista singolo linkata, ordinata e senza sentinelle. Le funzioni disponibili creano una nuova lista (`LISTtitoliInit`), inserisco in ordine un titolo (`LISTtitoliInsert`), cercano un titolo per codice (chiave di ricerca) (`LISTtitoliSearch`), memorizzano la lista (`LISTtitoliStore`) e verificano se è vuota (`LISTtitoliEmpty`)
- **Quasi ADT Quotazione:** struct con campo di tipo `Data` e 2 campi numerici per calcolare la quotazione del titolo come media di tutti i valori di quel titolo in quella data pesati sul numero di titoli scambiati. Le funzioni permettono di creare un nuovo oggetto vuoto (`QUOTAZIONEsetNull`), di verificare se un oggetto è vuoto (`QUOTAZIONEcheckNull`), di visualizzare un oggetto (`QUOTAZIONEstore`) e di estrarne il valore o la data (`QUOTAZIONEgetValore`, `QUOTAZIONEgetData`)
- **ADT di I classe** per la collezione di quotazioni implementato come BST: sono disponibili le funzioni standard di inizializzazione (`BSTquotazioniInit`), memorizzazione (`BSTquotazioniStore`), inserzione in foglia (`BSTquotazioniInsert`), ricerca (`BSTquotazioniSearch`) e verifica se vuoto (`BSTquotazioniEmpty`). Si sviluppano 3 funzioni specifiche per il problema:
 - `BSTgetMinMax`: si tratta di un'estensione della funzione standard di ricerca del minimo e del massimo in un BST adattata per ritornare tali informazioni in parametri passati by reference
 - `BSTquotazioniMinmaxRange`: la funzione ricorsiva `treeMinmaxRange` in funzione del confronto tra data corrente (chiave del BST) e date dell'intervallo scende nel sottoalbero sinistro o destro. Se invece la data corrente è compresa nell'intervallo, essa provvede ad aggiornare il massimo e il minimo valore della quotazione



- **BST** *quotazioni* *Balance*: il bilanciamento si realizza individuando la chiave mediana inferiore (la chiave mediana (inferiore) di un insieme di n elementi è l'elemento che si trova in posizione $\lfloor (n + 1)/2 \rfloor$ nella sequenza ordinata degli elementi dell'insieme), partizionando l'albero attorno ad essa con la funzione `treePartition` e ricorrendo sui sottoalberi sinistro e destro. Si osservi che esistono algoritmi più raffinati per il bilanciamento, quali ad esempio quello di Day, Stout e Warren, che, con complessità lineare, generano un albero binario quasi completo (tutti i livelli completi, eccetto l'ultimo, riempito da sinistra a destra, come uno heap). La funzione di bilanciamento è chiamata quando il rapporto tra lunghezza del cammino massimo radice-foglia (altezza dell'albero) e del cammino minimo radice-foglia eccede una soglia S . Detto rapporto è calcolato dalla funzione `treeMinmaxHeight` che esplora tutti i cammini radice-foglia tenendo traccia delle loro lunghezze al fine di identificare la minima e la massima.

Algoritmo: il `main` inizializza le strutture dati, legge 3 file di prova e offre un menu con le operazioni richieste dalle specifiche.