Università Degli Studi dell'Aquila

III appello del modulo di Laboratorio di Algoritmi e Strutture Dati A.A. 2023/2024
Martedì 13 febbraio 2024 - Dott.ssa Giovanna Melideo

Svolgere i seguenti esercizi avendo come riferimento il linguaggio JAVA.

Esercizio 1 Implementare una classe `Discendenza` che rappresenta relazioni di paternità tramite una mappa ordinata secondo l'ordinamento lessicografico definito sulle chiavi, che associa al codice fiscale di una persona il codice fiscale di suo padre.

Si implementino i seguenti metodi:

i. `String aggiungiRelazione(String CFfiglio, String CFpadre)` che effettua un inserimento o un aggiornamento dell'associazione CFfiglio-CFpadre e restituisce il precedente CF del padre associato a CFfiglio, se presente, o null se non è presente nessuna precedente associazione per CFfiglio;
ii. `String cancellaRelazione(String CFfiglio)` che cancella l'associazione per CFfiglio, se presente, e restituisce il precedente valore associato a CFfiglio o null se non è presente nessuna associazione per CFfiglio;
iii. `Set<String> getFigli()` che restituisce l'insieme dei CF di tutti i figli in ordine lessicografico crescente;
iv. `Set<String> getPadri()` che restituisce l'insieme dei CF di tutti i padri in ordine crescente rispetto al mese di nascita, rappresentato dal 9° carattere del CF ('A' = gennaio, 'B'= febbraio, ecc.);
v. `Set<String> getAscendenti(String CFfiglio)` che restituisce l'insieme dei CF degli ascendenti (padre, nonno, bisnonno, …) della persona specificata dal CF in input, se presente, o null se non è presente nessuna associazione per CFfiglio;
vi. `double AverageNumFigli()` che restituisce il numero medio di figli per padre presente nella mappa.

Esercizio 2 Realizzare un metodo statico

`public static boolean FatherIsSum(BinaryNode<Integer> root)` che, dato l'albero binario radicato nel nodo radice in input, verifica se ogni nodo interno (con almeno un figlio) contiene un intero che è la somma degli interi contenuti nei nodi figli.

Esercizio 3 Implementare un metodo costruttore `public Network(Vertex[] V)` della classe Network che prende in input i vertici (distinti) del digrafo e inserisce nel nuovo grafo un arco diretto da un vertice V[i] a un vertice V[j] (i ≠j) con probabilità ½, assegnando peso -1.0 se V[i]<V[j], 1.0 altrimenti.

Esercizio 4 Disegnare l'albero di ricerca 2-3-4 bilanciato risultante dall'inserimento della sequenza di chiavi `H, O, T, B, A, K, E, R, Y, S, U, N` (in questo ordine) in un albero inizialmente vuoto, usando il metodo di inserimento top-down. Trasformare l'albero risultante in un albero red-black.