

Programmazione ad Oggetti - parte B: Esercitazione di laboratorio 9

Esercizio 1: Leggere un file "info.dat" che contiene riga per riga un numero (ID) e una stringa.

Sviluppare un programma in C++ che svolga le seguenti operazioni:

1. memorizzare le informazioni lette dal file in un albero binario di ricerca utilizzando l'ID come chiave
2. leggere da tastiera nuove coppie di ID e stringa che vengono memorizzate in un secondo albero binario di ricerca avente lo stesso tipo di nodo del primo (questa fase termina se viene inserito un ID negativo)
3. eseguire il bilanciamento dei due alberi binari di ricerca e stamparli in logica post-order
4. scrivere un metodo "merge" della classe bst per recuperare i dati inseriti nei due alberi e inserirli in una lista singolarmente concatenata ordinata per ID crescente; qualora nei due alberi siano presenti nodi con lo stesso ID questi devono dare origine ad un singolo nodo nella lista ottenuto concatenando le stringhe con lo stesso ID
5. stampi il contenuto della lista.

Esercizio 2: Il file telefoni.txt contiene riga per riga: 1 un identificativo ufficio (stringa), nome e cognome di una persona (stringa) e un numero di telefono tutti separati dal carattere ",".

Esempio:

pal4,SAIA Francisco,905819

palA,ZENNA Michele,905751

pal1,BERTOZZI Massimo,905845

Scrivere un programma C++ che

1. legga il contenuto del file e lo memorizzi in due alberi binari di ricerca tali che:
 - i. il primo albero contenga come chiave il nome-cognome della persona e come valore l'ufficio
 - ii. il secondo albero contenga come chiave il numero di telefono e come valore il nome-cognome della persona
2. Stampi a video il contenuto dei due alberi in logica in-order dopo averli bilanciati
3. Stampi a video su ogni riga l'ufficio, il nome-cognome e il telefono di ogni persona (scrivere un algoritmo non efficiente basato su una visita del secondo albero utilizzando iterativamente la funzione "select")
4. Iterativamente chieda all'utente una stringa e
 - i. se la stringa è numerica cerchi il relativo numero di telefono e stampi, se la ricerca ha esito positivo, tutti i dati della persona corrispondente (altrimenti stampare un messaggio di errore)
 - ii. se la stringa non è numerica cerchi il nome inserito e stampi, se la ricerca ha esito positivo, **tutti** i nomi delle persone che lavorano nello stesso ufficio della persona cercata (altrimenti stampare un messaggio di errore). Nella ricerca dei nomi tenere

conto anche delle differenze tra caratteri maiuscoli e minuscoli (es: "PAOLO FEDERICI" è un nome diverso da "paolo Federici"). E' consentito l'uso della funzione select.

Esercizio 3:

Nei file "parteA.txt" e "parteB.txt" sono riportati i risultati di due prove di laboratorio di fondamenti di programmazione. Riga per riga i file riportano i dati di uno studente: matricola, cognome nome, voto, email separati da virgole. Ad esempio:

150755,Aldreggetti Ivano,22,ivano.aldreggetti@studenti.unipr.it

"cognome nome" e mail sono da considerarsi stringhe, matricola e voto sono numeri interi. Una votazione insufficiente viene riportata nel file con la stringa "INSUFFICIENTE" e deve essere convertita in un valore intero pari a 0.

Scrivere un programma che:

1. legga entrambi i file e inserisca tutti i dati in due distinti alberi binari di ricerca la cui chiave deve essere la matricola, li bilanci e li stampi in logica pre-order
2. effettui un attraversamento di uno dei due alberi e per ogni nodo visitato:
 - i. ricerchi la matricola corrispondente nel secondo albero
 - ii. calcoli il risultato complessivo dei due esami dello studente come media algebrica dei due risultati assegnando a INSUFFICIENTE il valore 0
 - iii. stampi il valore medio se sufficiente (≥ 18) oppure INS se insufficiente (< 18)
3. stampi la media dei voti sufficienti per ciascuna delle due prove