



Esercitazione di laboratorio n. 11

(Caricamento nella Sezione Elaborati del Portale entro e non oltre il 23/12/2014)

Esercizio n. 1: ADT di I Categoria - BST

Si realizzi un programma C che, attraverso un'apposita interfaccia utente, permetta di gestire una struttura dati di tipo BST (albero binario di ricerca).

Le operazioni permesse devono essere quelle di:

- creazione di un nuovo BST (vuoto)
- inserimento in foglia di un nuovo elemento nel BST
- inserimento in radice di un nuovo elemento nel BST
- ricerca di un elemento (specificato dall'utente) nel BST
- stampa dell'elemento minimo e massimo presenti nel BST
- visualizzazione (a video) di tutti gli elementi presenti nel BST, alternativamente secondo pre/post/in order
- calcolo dell'altezza dell'albero
- calcolo del numero totale di nodi nell'albero
- calcolo del numero di nodi foglia
- salvataggio del BST su file (elencando i nodi in un ordine qualsiasi)
- caricamento di un nuovo BST da file (vedi esempio di file a seguire) mediante una sequenza di inserzioni in foglia. Ogni riga del file contiene le informazioni relative al nodo da inserire nella forma intero – stringa.

Il programma deve essere realizzato su tre moduli distinti:

- l'interfaccia utente
- un modulo per l'ADT di prima categoria BST
- un modulo per l'ADT di prima categoria del tipo Item, da usarsi come informazione memorizzata nei nodi dell'albero. Sia l'Item composto da due campi: un intero e una stringa di lunghezza massima pari a 25 caratteri.

Sia il campo intero dell'Item la chiave su cui basare l'ordinamento del BST.

Esempio di file di input:

```
3 nodo1
4 nodo2
1 nodo3
6 nodo4
```

Esercizio n. 2: Tabelle di simboli

Realizzare un programma in linguaggio C per la gestione delle statistiche di utilizzo dei servizi di una compagnia che offre contenuti multimediali in streaming.

Un primo file (file “clienti.txt”) contiene la traccia dei clienti della compagnia. Ogni riga del file ha il seguente formato:

<idCliente> <Cognome> <Nome> <Categoria_Cliente>

dove i vari campi sono stringhe di caratteri alfanumerici indicanti l'identificatore del cliente, i suoi dati identificativi e la categoria a cui appartiene. Tutte le stringhe sono lunghe al massimo 25 caratteri.



Un secondo file (file "traffico.txt") contiene la traccia degli accessi a servizi per il mese corrente. Ogni riga del file ha il seguente formato:

`<idCliente> <QuantitaKB> <DurataConnessione>`

L'identificatore del cliente segue le stesse regole di cui sopra e il medesimo id può apparire più volte all'interno del secondo file. QuantitaKB e DurataConnessione sono due interi, a rappresentare rispettivamente il numero di KB trasferiti e la durata, in secondi, di ogni connessione.

Si noti che tanto il numero di accessi quanto quello di clienti è ignoto; per entrambi i file non si può supporre nessun ordine particolare nella memorizzazione delle informazioni.

L'applicazione deve essere in grado di determinare e memorizzare su file:

- la quantità totale di dati trasmessi per categoria di cliente
- i clienti suddivisi per ciascuna categoria
- il numero di accessi effettuati nonché la quantità di dati trasferiti e la durata totale della connessione per ciascun cliente.

Tali informazioni dovranno essere salvate su tre file distinti con il formato riportato nell'esempio.

I nomi dei cinque file (due di ingresso e tre di uscita) siano ricevuti quali parametri sulla riga di comando.

I dati letti dal primo file siano organizzati in una tabella di simboli basata su tabelle di hash con chiave `<idCliente>` di modo che, dato `<idCliente>`, si abbia accesso diretto a tutti i dati del cliente.

Sono possibili 2 alternative per la tabella di hash:

1. implementare un ADT di I categoria
2. interfacciarsi all'ADT ST presentato a lezione e disponibile sul Portale.