

# Progetto di Algoritmi e Strutture Dati - A.A. 2014-15

Versione 0.5  
del 15/01/2015

## Esercizio 1

L'associazione "Amici dell'algoritmica" ha necessità di gestire una collezione di valori di tipo numerico ognuno dei quali identifica in maniera univoca un proprio socio.

In ogni momento nuovi soci si possono iscrivere all'associazione così come possono revocare la propria iscrizione. Ogni volta che una persona diventa socio il suo identificativo viene aggiunto alla collezione; analogamente ogni volta che un socio non rinnova la propria iscrizione l'identificativo viene rimosso dalla collezione.

La struttura dati astratta che gestisce gli identificativi dei soci dovrà essere una lista. Si forniscano due implementazioni di tale struttura dati, una che fa uso di un array la cui dimensione varia dinamicamente (struttura dati 1), l'altra che fa uso di variabili di tipo reference per collegare gli elementi della lista (struttura dati 2).

Ogni operazione di inserimento e di cancellazione di un elemento dovrà poi essere eseguita in maniera analoga su entrambe le implementazioni. Dopo ogni operazione occorre individuare sperimentalmente la memoria effettivamente occupata da ogni struttura dati.

### Note implementative

Il file che contiene il main si dovrà chiamare Esercizio1.java.

I dati da inserire dovranno essere letti dal file Input1Es1.txt che conterrà righe secondo il seguente formato:

I *numero*  
R *numero*

Le righe che iniziano con I indicano che il *numero* dovrà essere inserito nella struttura dati, le righe che iniziano con R che il numero dovrà essere rimosso. (ovviamente le operazioni di inserimento e cancellazione potranno succedersi in maniera random)

Dopo ogni operazione di inserimento o cancellazione dovrà essere stampata sul file Output1Es1.txt l'effettiva occupazione di memoria in termini di byte di entrambe le strutture dati secondo questo formato:

*occupazione1 occupazione2*

dove i valori *occupazione1* e *occupazione2* rappresentano rispettivamente l'occupazione della memoria della struttura dati 1 e 2.

Il file Input1Es1.txt sarà messo a disposizione sul sito del corso.

Si esegua il programma anche sul file Input2Es1.txt. Tale file, preparato dallo studente, dovrà avere lo stesso formato del precedente e dovrà contenere almeno un migliaio di righe con una prevalenza di operazioni di inserimento nelle prime 200 righe.

Analogamente si stampino i risultati inerenti all'occupazione di memoria sul file Output2Es1.txt .

Nel main dovranno quindi essere svolte le seguenti operazioni:

- lettura del file contenente le operazioni da effettuare
- esecuzione delle operazioni in esse descritte sulle due strutture dati
- stampa dell'occupazione di memoria sul file di output

I nomi dei file utilizzati dovranno essere inseriti come parametri a riga di comando. Ovvero il programma dovrà essere eseguito nel seguente modo:

```
java Esercizio1 Input1Es1.txt Output1Es1.txt
```

e

```
java Esercizio1 Input2Es1.txt Output2Es1.txt
```

Una volta ottenuti i dati di output fare un'analisi degli stessi confrontando l'occupazione di memoria della struttura dati 1 con quelli della struttura dati 2.

## **Esercizio 2**

Si scriva un programma che implementi la costruzione di un albero binario di ricerca a partire da un insieme di valori dati in input. Ogni nodo dell'albero dovrà contenere solo il puntatore al padre e i puntatori ai figli (oltre naturalmente al valore del nodo). Si implementino poi la pre-visita, l'in-visita e la post-visita dell'albero in maniera iterativa.

### Note implementative

Il file che contiene il main si dovrà chiamare Esercizio2.java.

I valori da inserire all'interno dell'albero binario di ricerca dovranno essere letti dal file InputEs2.txt. e che verrà messo a disposizione sul sito del corso.

Il risultato della pre-visita, invisita e postvisita dovrà essere stampato nel file OutputEs2.txt (un valore per ogni riga, tra una visita e l'altra dovranno essere lasciate due righe vuote)

I nomi dei file utilizzati dovranno essere inseriti come parametri a riga di comando. Ovvero il programma dovrà essere eseguito nel seguente modo:

```
java Esercizio2 InputEs2.txt OutputEs2.txt
```

### **Esercizio 3**

La biblioteca “Leggi un libro” ha necessità di memorizzare e gestire, per ogni libro, varie informazioni tra cui il titolo, gli autori, l’anno di pubblicazione, il numero di copie possedute. Per ogni copia del libro posseduta, inoltre, occorre memorizzare la collocazione e i dati relativi al prestito. Un libro è identificato univocamente da un codice ISBN, ma per ogni libro possono essere presenti più copie. Pur non essendoci limite al numero di copie per ogni libro, si consideri che raramente saranno presenti più di 5 copie di uno stesso libro.

Implementare un programma che permetta di gestire i vari libri posseduti dalla biblioteca. Tale programma dovrà permettere di inserire nuovi libri nell’archivio, eliminare le copie dei libri destinati al macero, modificare le informazioni relative al prestito.

Si forniscano due implementazioni del programma, ognuna dovrà utilizzare la struttura dati più opportuna in termini di complessità computazionale: nella prima implementazione (Esercizio3a) di supponga che le operazioni di ricerca siano molto più comuni rispetto all’inserimento e alla cancellazione, nella seconda che le operazioni di ricerca, inserimento e cancellazione presentino la stessa frequenza.

Per semplicità, si può supporre che la ricerca possa avvenire solo per codice ISBN.

Calcolare la complessità computazionale delle varie soluzioni adottate e confrontarla con i risultati sperimentali ottenuti.

#### Note implementative

Il file che contiene il main si dovrà chiamare Esercizio3a.java per la prima implementazione e Esercizio3b.java per la seconda.

I dati relativi ai libri dovranno essere letti per entrambi i programmi dal file InputEs3.txt che verrà messo a disposizione sul sito del corso e che avrà il seguente formato:

codice ISBN - titolo libro - autore - n.ro di copie

es.

8804525193 - I malavoglia - G. Verga - 3

Le altre informazioni relative al libro e non presenti nel file potranno essere tralasciate.

Le operazioni da eseguire per il programma Esercizio3a e Esercizio3b dovranno essere lette rispettivamente dai file InputEs3a.txt e InputEs3b.txt che conterranno righe secondo il seguente formato:

I - *codice ISBN - titolo libro - autore - n.ro di copie*

R - *codice ISBN*

F - *codice ISBN*

Le righe che iniziano con I indicano che il libro dovrà essere inserito nella struttura dati, le righe che iniziano con R che il libro dovrà essere rimosso e le righe che iniziano con F che il libro dovrà essere cercato (ovviamente le operazioni di inserimento, cancellazione e ricerca potranno succedersi in maniera random).

I file InputEs3a.txt e InputEs3b.txt saranno messi a disposizione sul sito del corso.

I programmi Esercizio3a e Esercizio3b dovranno produrre rispettivamente i file OutputEs3a.txt e OutputEs3b.txt. Ogni riga di questi file dovrà contenere il risultato dell'operazione indicata nella stessa riga del file di input. Per le operazioni di inserimento si scriverà semplicemente "Inserimento riuscito", per le operazioni di cancellazione "cancellazione riuscita", per le operazioni di ricerca il risultato della ricerca (es. 8804525193 - I malavoglia - G. Verga - 3) oppure "libro non trovato".

I nomi dei file utilizzati dovranno essere inseriti come parametri a riga di comando. Ovvero i programmi dovranno essere eseguiti nel seguente modo:

```
java Esercizio3a InputEs3.txt InputEs3a.txt OutputEs3a.txt
```

```
java Esercizio3b InputEs3.txt InputEs3b.txt OutputEs3b.txt
```

## **Esercizio 4**

Tutti i libri della biblioteca "Leggi un libro elettronico" sono memorizzati su supporti di memoria secondaria; in memoria principale vengono tenute le seguenti informazioni: codice ISBN, titolo del libro, autore/i e puntatore alla copia elettronica del libro presente su memoria secondaria. Quando viene richiesto un libro presente nella biblioteca, si accede, utilizzando il puntatore, alla copia elettronica, quest'ultima viene trasferita in memoria principale e inviata, per posta elettronica, al richiedente. Nel caso il libro richiesto non sia presente nella biblioteca, al richiedente viene inviato un messaggio di richiesta non soddisfatta.

Utilizzando una struttura dati AB-tree di grado  $b$ , realizzare un programma che permetta la gestione dei libri posseduti dalla biblioteca. Tale programma dovrà permettere oltre che la ricerca di uno specifico libro, anche l'inserimento di nuovi libri nella biblioteca e la cancellazione di libri.

Notare che un AB-Tree di grado  $b$  ha le seguenti proprietà

- ogni nodo  $v$  diverso dalla radice mantiene  $k(v)$  chiavi ordinate:  
 $chiave_1(v) \leq chiave_2(v) \leq \dots \leq chiave_{k(v)}(v)$  tali che  $1 \leq k(v) \leq b$
- la radice mantiene almeno 1 ed al più  $b$  chiavi ordinate.
- ogni nodo  $v$  ha  $k(v)+1$  figli

- le chiavi  $chiave_i(v)$  separano gli intervalli di chiavi memorizzati in ciascun sottoalbero. Se  $c_i$  è una qualunque chiave nell' $i$ -esimo sottoalbero di un nodo  $v$ , allora  $c_1 \leq chiave_1(v) \leq c_2 \leq chiave_2(v) \leq \dots \leq c_k(v) \leq chiave_{k(v)}(v) \leq c_{k(v)+1}$

Per semplicità, si può supporre che la ricerca possa avvenire solo per codice ISBN. Nel caso di cancellazione di un titolo, si assuma che l'eliminazione della chiave, rappresentata dal codice ISBN, sia sufficiente a rendere disponibile lo spazio di memoria secondaria su cui era memorizzato il libro. Si supponga, inoltre, che lo spazio di memoria secondaria disponibile sia sufficiente a contenere tutti i libri della biblioteca.

### Note implementative

Il file che contiene il main si dovrà chiamare `Esercizio4.java`.

Le operazioni da eseguire per il programma `Esercizio4` dovranno essere lette dai file `InputEs4.txt` e conterranno righe secondo il seguente formato:

*I - codice ISBN - titolo libro - autore - puntatore*  
*R - codice ISBN*  
*F - codice ISBN*

Ad esempio:

8804525193 - I malavoglia - G. Verga - 0xEA8902BCCC

Le righe che iniziano con I indicano che il libro dovrà essere inserito nella struttura dati, le righe che iniziano con R che il libro dovrà essere rimosso e le righe che iniziano con F che il libro dovrà essere cercato (ovviamente le operazioni di inserimento, cancellazione e ricerca potranno succedersi in maniera random).

Il file `InputEs4.txt` sarà messo a disposizione sul sito del corso.

Il programma `Esercizio4` dovrà produrre il file `OutputEs4.txt`. Ogni riga di questo file dovrà contenere il risultato dell'operazione indicata nella stessa riga del file di input. Per le operazioni di inserimento si scriverà semplicemente "Inserimento riuscito", per le operazioni di cancellazione "cancellazione riuscita", per le operazioni di ricerca il risultato della ricerca (es. 8804525193 - I malavoglia - G. Verga) oppure "libro non trovato".

Il grado  $b$  e i nomi dei file utilizzati dovranno essere inseriti come parametri a riga di comando.

Ovvero il programma dovrà essere eseguito nel seguente modo:

```
java Esercizio4 b InputEs4.txt OutputEs4.txt
```

dove  $b$  dovrà essere sostituito da un valore intero che rappresenta il grado dell'AB-albero implementato.

## Tempi e modalità di consegna

I file di input verranno pubblicati al seguente link: <http://www.cs.unibo.it/~turrini/DIDATTICA/ALGORITMI/ANNO1415/PROGETTO/> entro il 23 dicembre 2014.

Il progetto dovrà essere svolto singolarmente.

Si precisa che verrà anche valutato l'utilizzo corretto e opportuno del linguaggio Java. I file sorgenti dovranno essere opportunamente commentati.

Inoltre è richiesta la redazione di una relazione che dovrà contenere:

- la descrizione delle scelte progettuali attuate
- la stima della complessità computazionale ove richiesto
- i grafici opportunamente commentati ove richiesto

I sorgenti degli esercizi (file .java) e la documentazione dovrà essere posta in una directory chiamata con il cognome dello studente che ha svolto il progetto. Tale directory andrà poi compressa (formato zip o jar) e spedita a [lorenzo.donatiello@unibo.it](mailto:lorenzo.donatiello@unibo.it) e a [elisa.turrini7@unibo.it](mailto:elisa.turrini7@unibo.it) entro il 30 gennaio 2015. Per la spedizione è obbligatorio utilizzare l'indirizzo di email istituzionale.

Entro il 10 febbraio 2015 verrà reso noto chi è stato ammesso alla discussione del progetto, discussione che si terrà a partire dal giorno successivo.

Si ricorda che è possibile consegnare gli esercizi solo qualora questi siano funzionanti e rispettino le specifiche proposte. Per automatizzare la compilazione, le classi non dovranno essere contenute in pacchetti o suddivise per directory. La compilazione e l'esecuzione avverrà digitando da shell i seguenti comandi:

```
javac *.java
```

```
java Esercizio1 Input1Es1.txt Output1Es1.txt
```

```
java Esercizio1 Input2Es1.txt Output2Es1.txt
```

```
java Esercizio2 InputEs2.txt OutputEs2.txt
```

```
java Esercizio3a InputEs3.txt InputEs3a.txt OutputEs3a.txt
```

```
java Esercizio3b InputEs3.txt InputEs3b.txt OutputEs3b.txt
```

```
java Esercizio4 b InputEs4.txt OutputEs4.txt
```