

Algoritmi e Strutture Dati 1

Appello del 25/05/2011

Esercizio 1 (9 punti)

Scrivere un metodo `public static int sommaInsieme (int [] A)` in Java che preso in input un array A di numeri interi restituisce in output la somma degli elementi presenti in A, **contando una sola volta gli elementi ripetuti**. La procedura **NON può modificare** l'array A.

Ad esempio, se $A=[1,3,2,3,5,1,3]$, il metodo deve restituire $1+3+2+5=11$, senza contare più di una volta nella somma gli interi 1 e 3 che sono ripetuti.

La procedura deve avere tempo di esecuzione nel caso peggiore $O(n^2)$, e NON può sfruttare nessuna delle procedure viste a lezione.

Esercizio 2 (9 punti)

Scrivere un metodo `public static int sommaInsiemeBis (int [] A)` in Java che preso in input un array A di numeri interi restituisce in output la somma degli elementi presenti in A, **contando una sola volta gli elementi ripetuti**. La procedura **NON può modificare** l'array A.

Ad esempio, se $A=[1,3,2,3,5,1,3]$, il metodo deve restituire $1+3+2+5=11$, senza contare più di una volta nella somma gli interi 1 e 3 che sono ripetuti.

La procedura deve avere tempo di esecuzione nel caso peggiore $O(n \log n)$, e può sfruttare tutte le procedure viste a lezione (es: *InsertionSort, MergeSort, Quicksort, Binarysearch, LinearSearch, etc...*)

Esercizio 3 (9 punti)

Si consideri la classe SimpleList

```
public class SimpleList{
    int key;
    SimpleList next;
    ...
}
```

che rappresenta un elemento di una lista e possiede una chiave intera ed il puntatore all'elemento successivo. Si noti come una lista è identificata in modo naturale dal suo elemento di testa.

Scrivere un metodo **ricorsivo** `public static boolean ricerca (SimpleList L, int k)` in Java che data una SimpleList L ed una chiave k restituisce **true** se e solo se k è presente nella lista L.

Esercizio 4 (3 punti + 6 punti)

a. Data la seguente sequenza di chiavi intere:

- **24, 8, 74, 31, 12, 9, 22, 10, 18, 99**

mostrare, a partire da strutture dati vuote, ed inserendo nell'ordine indicato le chiavi su elencate

1. l'**albero binario di ricerca** che si ottiene
2. l'**heap di minimo** che si ottiene
3. la **tabella Hash** di **dimensione 11** con **liste di trabocco** e funzione hash $H(x) = x \bmod 11$ che si ottiene

b. Mostrare l'ordine di visita dei nodi dell'albero costruito al punto (a.1) rispetto ai seguenti algoritmi di visita:

- Visita in ordine intermedio
- Visita in preordine
- Visita in postordine

Attenzione:

- Per svolgere il compito si hanno a disposizione **90** minuti.
- Scrivere **subito** nome, cognome, matricola e numero del compito su **OGNI FOGLIO**.
- Durante la prova scritta non è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**