

Programmazione e Algoritmi 1

A.A. 2023/24 — Compito dell'8 luglio 2024 – Versione Java

prof. Gianluca Amato

Gli esercizi di programmazione saranno valutati sulla base della correttezza, efficienza e comprensibilità della soluzione proposta. In generale, **se volete usare una funzione o un metodo che non è stato presentato a lezione, chiedete prima al docente se è consentito.**

Esercizio 1 (5 punti)

Descrivere l'algoritmo di ordinamento per fusione (merge sort): input, output, funzionamento, vantaggi e svantaggi rispetto ad altri algoritmi che risolvono lo stesso problema. Successivamente, illustrarne il funzionamento applicandolo all'array {"ciriciao", "a", "tutti", "sono", "arale"}.

Esercizio 2 (8 punti)

Si consideri il seguente codice Java, e se ne scriva la traccia di esecuzione negli appositi moduli.

```
1  import java.util.Arrays;
2
3  public class Esercizio2 {
4
5      static void mistero(int[] l1, int[] l2) {
6          for (int i = 0; i < l1.length; i++) {
7              l1[i] += mistero2(l2, i);
8          }
9      }
10
11     static int mistero2(int[] l, int i) {
12         int x = 0;
13         while (i >= 0) {
14             x += l[i];
15             i -= 1;
16         }
17         return x;
18     }
19     public static void main(String[] args) {
20         int[] data1 = { 1, 2 };
21         int[] data2 = { 2, 3, 4, 5 };
22         mistero(data1, data2);
23         System.out.println(Arrays.toString(data1));
24     }
25 }
```

Esercizio 3 (5 punti)

Scrivere un metodo statico `estraiMaschera(int[] data, boolean[] maschera)`. Il metodo, senza modificare i parametri in input, restituisce un nuovo array che contiene gli elementi di `data` che corrispondono a valori `true` in `maschera`. Ad esempio `estraiMaschera(new int[] {2, 3, 5, 7}, new boolean[] {true, false, false, true})` restituisce l'array {2, 7}.

Se la lunghezza di `maschera` è minore di quella di `data`, gli elementi mancanti vengono considerati uguali a `false`. Se la lunghezza di `maschera` è maggiore di quella di `data`, gli elementi in più vengono semplicemente ignorati.

Esercizio 4 (5 punti)

Scrivere due test nel framework `junit` per verificare il corretto funzionamento di `estraiMaschera`:

- un test per le seguenti chiamate di metodo:
 - `estraiMaschera(new int[] {2, 3, 5, 7}, new boolean[] {true, false, false, true})`
 - `estraiMaschera(new int[] {2, 3, 5, 7}, new boolean[] {true})`
 - `estraiMaschera(new int[] {2, 3}, new boolean[] {false, true, true})`
- un test per il caso in cui l'array `data` è generato casualmente (sia la lunghezza che il contenuto sono casuali) e `maschera` è sempre `{true}`; questo test deve essere ripetuto 100 volte.

Esercizio 5 (5 punti)

Scrivere il metodo `estraiMaschera2` che modifica `estraiMaschera` in modo che, se la lunghezza di `maschera` è minore di quella di `data`, l'array `maschera` venga ripetuto tante volte quanto è necessario per arrivare alla lunghezza di `data`. Ad esempio

```
estraiMaschera2(new int[] {2, 3, 5, 7}, new boolean[] {true, false})
```

restituisce l'array `{2, 5}`, perché la `maschera` viene considerata come fosse `{true, false, true, false}`. Stessa cosa, in maniera simmetrica, se è `data` a essere più corta di `maschera`. Pertanto

```
estraiMaschera2(new int[] {2, 3}, new boolean[] {false, true, true, false})
```

restituisce l'array `{3, 2}`, perché è come se l'array `data` fosse `{2, 3, 2, 3}`.

Si può supporre che le lunghezze di `data` e `maschera` siano non nulle, ed una un multipla dell'altra.

Esercizio 6 (5 punti)

Si consider il seguente gioco per due giocatori. Sul tavolo c'è un certo numero di fiammiferi. Ogni giocatore, a turno, può prendere 1, 2 o 3 fiammiferi alla volta. Vince chi prende l'ultimo fiammifero. Scrivere un programma Java che implementi questo gioco. Il programma chiede all'utente di inserire il numero di fiammiferi iniziale, e poi fa giocare i due giocatori a turno. Il programma deve controllare che il numero di fiammiferi presi sia corretto. Il programma deve stampare a video il numero di fiammiferi rimasti dopo ogni mossa, e alla fine deve stampare il nome del giocatore che ha vinto. Questo è un esempio di interazione: in rosso quello scritto dall'utente, in nero l'output del programma.

```
Con quanti fiammiferi vuoi iniziare ? 7
Ora ci sono 7 fiammiferi.
Giocatore 1, quanti fiammiferi vuoi prendere (1, 2 o 3)? 7
Risposta non valida
Ora ci sono 7 fiammiferi.
Giocatore 1, quanti fiammiferi vuoi prendere (1, 2 o 3)? 3
Ora ci sono 4 fiammiferi.
Giocatore 2, quanti fiammiferi vuoi prendere (1, 2 o 3)? 3
Ora ci sono 1 fiammiferi.
Giocatore 1, quanti fiammiferi vuoi prendere (1, 2 o 3)? 3
Risposta non valida
Giocatore 1, quanti fiammiferi vuoi prendere (1, 2 o 3)? 1
Ha vinto il giocatore 1.
```