

Programmazione e Algoritmi 1

A.A. 2023/24 — Compito del 22 gennaio 2024 – Versione Java

prof. Gianluca Amato

Gli esercizi di programmazione saranno valutati sulla base della correttezza, efficienza e comprensibilità della soluzione proposta. In generale, **non è possibile usare classi che non fanno parte del programma del corso**.

Esercizio 1 (4 punti)

Scrivere un metodo statico `sameSet` che prende in input due array e determina se essi hanno gli stessi elementi, ignorando sia l'ordine che i duplicati. Per esempio, gli array `{1, 4, 9, 16, 9, 7, 4, 9, 11}` e `{11, 11, 7, 9, 16, 4, 1}` hanno gli stessi elementi, mentre `{1, 4, 9, 19}` e `{4, 4, 19, 1, 3}` no. La funzione deve restituire `true` se i due array hanno gli stessi elementi, `false` altrimenti.

Esercizio 1bis (4 punti)

Si scriva una variante del metodo `sameSet` dell'Esercizio 1 specificatamente per il caso in cui i due array sono ordinati. La variante deve avere complessità $O(n)$, dove n è la lunghezza dell'array più lungo.

Esercizio 2 (4 punti)

Scrivere alcuni test nel framework JUnit per verificare il corretto funzionamento di `sameSet`. In particolare:

- controllare il corretto funzionamento della funzione per gli input di esempio dell'Esercizio 1;
- controllare il corretto funzionamento della funzione quando un array è generata casualmente, e il secondo è ottenuto dal primo scambiando casualmente due elementi.

Esercizio 3 (4 punti)

Un quadrato magico è una matrice quadrata di numeri interi positivi tale che la somma dei numeri di ogni riga, di ogni colonna e delle due diagonali è uguale. Per esempio, la seguente matrice è un quadrato magico:

2	7	6
9	5	1
4	3	8

Tutte le righe, le colonne e le diagonali hanno somma 15.

Scrivere un metodo static `isMagicSquare` che prende in input una matrice quadrata di numeri interi e restituisce `true` se la matrice è un quadrato magico, `false` altrimenti. Ad esempio `isMagicSquare(new int[][]{{2, 7, 6}, {9, 5, 1}, {4, 3, 8}})` deve restituire `true`.

Esercizio 4 (4 punti)

Scrivere un programma `MagicSquare` che legge dalla riga di comando una matrice quadrata di numeri interi, e determina se si tratta o meno di un quadrato magico (è ovviamente possibile usare la funzione `isMagicSquare`). Il primo parametro della riga di comando è la dimensione della matrice, mentre i restanti sono i valori della matrice, letti per riga. Ad esempio, se lanciato con il comando

```
java MagicSquare 3 2 7 6 9 5 1 4 3 8
```

il programma deve stampare il fatto che si tratti di un quadrato magico.

Esercizio 5 (5 punti)

Illustrare il funzionamento dell'algoritmo di ordinamento per selezione (selection sort) prima in termini generali, e poi applicandolo all'array `{3, 5, 2, 1, 4}`. Discutere la complessità computazionale dell'algoritmo.

Esercizio 6 (8 punti)

Si consideri il seguente codice Java:

```
1  class ProgJava {
2
3      static boolean f1(int x, int i, int[] l) {
4          for (int j = i; j < l.length; j++)
5              if (l[j] == x)
6                  return true;
7          return false;
8      }
9
10     static boolean f2(int[] l) {
11         for (int i = 0; i < l.length; i++)
12             if (f1(l[i], i+1, l))
13                 return true;
14         return false;
15     }
16     public static void main(String[] args) {
17         System.out.println(f2(new int[]{1, 2, 5, 3, 5}));
18     }
19 }
```

Eseguire passo passo il programma (senza l'ausilio di un computer) e fornire la traccia di esecuzione. **Inoltre**, spiegare a parole cosa fanno le funzioni `f1` ed `f2`.