

Appello di Programmazione e Algoritmi 1
9 febbraio 2021

Esercizio 1

Cosa stampa il seguente frammento di codice Java?

```
int [] A = {9,2,6,7};
int [] B = {6,9,6,2};
int conto=0;
for (int i=A.length-1;i>=0;i--) {
    if (A[i]>B[i]){
        System.out.println(A[i]-B[i]);
        conto++;
    }
}
System.out.println(conto);
```

NON SARANNO CONSIDERATE VALIDE LE RISPOSTE NON GIUSTIFICATE DAI CALCOLI EFFETTUATI PER OTTENERE IL RISULTATO

Cosa stampa il seguente programma Java?

```
public static void main(String[] args) {
    System.out.println(enigma(1));
    System.out.println(enigma(5));
    System.out.println(enigma(8));
}
public static int enigma (int x){
    if (x<1) return 0;
    if (x%2==1) return x + enigma(x+1);
    return x+2 + enigma (x-3);
}
```

NON SARANNO CONSIDERATE VALIDE LE RISPOSTE NON GIUSTIFICATE DAI CALCOLI o DAI RAGIONAMENTI EFFETTUATI PER OTTENERE IL RISULTATO

Esercizio 2

Scrivere un metodo iterativo

public static int[] noUguali (int[] a)

che, preso come parametro l'array **a** di numeri interi restituisce un nuovo array ottenuto da **a** cancellando tutti gli elementi adiacenti che hanno valori identici, mantenendo l'ordine iniziale degli altri elementi. Ad esempio, se $a = \{6, 6, 6, 2, 3, 1, 1, 7, 7, 4, 7\}$ allora il risultato è $\{2, 3, 4, 7\}$ (non vengono inseriti i valori uguali a 6 nelle posizioni di indice 0,1 e 2, i valori uguali a 1 nelle posizioni di indice 5 e 6, i valori uguali a 7 nelle posizioni di indice 7 e 8).

Altro esempio: se $a = \{6, 2, 6, 6, 3, 1, 1, 3\}$ allora il risultato è $\{6, 2, 3, 3\}$

Si analizzi la complessità temporale del metodo proposto (soluzioni con complessità minore verranno valutate maggiormente).

Esercizio 3

Scrivere un metodo

static int[] ripeti (int[] a, int k)

che, presi come parametri un array **a** di numeri interi e un numero intero **k** restituisce un nuovo array ottenuto ripetendo **k** copie di **a**. Ad esempio se **a**={1, 2, 2} e **k**=3 dovrà essere restituito l'array {1, 2, 2, 1, 2, 2, 1, 2, 2} (ovvero un array che ha come elementi gli elementi dell'array **a** ripetuti per 3 volte, nell'ordine in cui compaiono). Se **a** è null o **k** è minore di 0 deve essere restituito null. Se **k** è 0 dovrà essere restituito l'array vuoto.

Altri esempi, se **a**={1, 4, 5, 2} e **k**=1 viene restituito l'array {1, 4, 5, 2}.

Se **a**={1, 4, 5, 5} e **k**=2 viene restituito {1, 4, 5, 5, 1, 4, 5, 5}

Si analizzi la complessità temporale del metodo proposto.

Esercizio 4

Si consideri il seguente array di numeri interi:

{12,6,33,21,16,11,7,9,3,10}

Mostrare **passo-passo l'esecuzione del merge-sort** per ordinare l'array in modo non decrescente.

Si descriva la complessità computazionale del merge-sort.

[Esercizio di Laboratorio] In relazione al progetto di laboratorio, si implementino i seguenti due metodi:

- int ultimaFigurina(boolean[] album)

restituisce il numero dell'ultima figurina presente nell'album (quella di indice maggiore). Se l'album è vuoto restituisce -1.

- boolean[] creaAlbum(int n, int[] figurine)

restituisce un nuovo album di lunghezza **n** che contenga le figurine i cui indici sono presenti nell'array **figurine**. Ad esempio, con input **n**=7 e l'array {2,3,5} il metodo restituisce l'album:
{false, false, true, true, false, true, false};

Non è necessario implementare il metodo main, ma solamente i due metodi richiesti.

Nell'implementazione dei metodi è possibile sfruttare altri metodi del progetto, se lo ritenete utile.

