Appello di Programmazione e Algoritmi 1
23 febbraio 2021

Esercizio 1

Cosa stampa il seguente frammento di codice Java?

```
int [] a= {4,3,6,3,7};
int conta =a.length;
for (int i = 0; i < a.length-1; i++) {
     for (int j = i+1; j < a.length -i; j++) {
          if (a[i] <= a[j]) System.out.println(a[i]+a[j]);
          else conta--;
          }
     }
System.out.println(conta);
```

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(prova(4));
          System.out.println(prova(0));
          System.out.println(prova(9));
          }

     static int prova (int x){
          if (x<0) return x;
          if (x%2==0) return x+prova(x-1);
          return x+prova(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 boolean moltiplicaIVicini (int[] a)
che, preso come parametro l'array **a** di numeri interi restituisce true se l'array contiene almeno un elemento (esclusi il primo e l'ultimo) che sia uguale al prodotto dell'elemento che lo precede e di quello che lo segue. In tal caso il programma stampa anche l'indice dell'elemento trovato. Il programma restituisce false altrimenti.
Ad esempio se a={ 8, 16, 2, 6, 11, 5, 10, 5, 17} il metodo restituisce true e stampa 1 poiché a[1]=16 ed è uguale al prodotto di a[0] e di a[2]. Se a={ 8, 15, 2, 6, 11, 5, 10, 5, 17} il metodo restituirà false.
Si analizzi la complessità temporale del metodo proposto (soluzioni con complessità minore verranno valutate maggiormente).

```
public static boolean moltiplicaIVicini (int[] a)
     {if (a==null|| a.length<=2) return false;
     for (int i=1; i<a.length-1; i++)
          if (a[i]==a[i-1]*a[i+1])
               {System.out.println(i); return true;}
     return false;}
```

complessità **θ(n) dove n**=a.length

Esercizio 3
Scrivere un metodo
static int[] ripeti (int[] a, int k)
che, preso come parametro l'array **a** di numeri interi ed un intero **k**, restituisce un nuovo array ottenuto ripetendo k volte tutti gli elementi divisibili per k stesso e lasciando inalterati tutti gli altri elementi. Ad esempio, se a={6,1,9,1,4,6} e k=3 allora dovrà essere restituito il nuovo array {6,6,6,1,9,9,9,1,4,6,6,6} (il 6, nella prima e nell'ultima posizione di a, e il 9 vengono riscritti 3 volte). Altro esempio, se a={6, 5} e k=7, dovrà essere restituito il nuovo array {6, 5}.
Si analizzi la complessità temporale del metodo proposto (soluzioni con complessità minore verranno valutate maggiormente).

DUE POSSIBILI SOLUZIONI

```
static int[] ripeti (int[] a, int k)
      {if (a==null || k<=0) return null;
      int conta=a.length;
      for (int i=0; i<a.length; i++)
      {if (a[i]%k==0) conta=conta+k-1;}
      System.out.println(conta);
      int[] result= new int [conta];
      int l=0;
      for (int i=0; i<a.length; i++)
            { result[l]=a[i]; l++;  //a[i] si copia una volta
            if (a[i]%k==0)
      // se la condizione dell'if è vera, a[i] si ricopia altre k-1 volte
               {for (int j=0; j<k-1; j++)
                   {result[l]=a[i]; l++;}
               }
            }
     return result;}

      static int[] ripeti2 (int[] a, int k)
      {if (a==null || k<=0) return null;
      int conta=a.length;
      for (int i=0; i<a.length; i++)
      {if (a[i]%k==0) conta=conta+k-1;}
      System.out.println(conta);
      int[] result= new int [conta];
      int l=0;
      for (int i=0; i<a.length; i++)
            {if (a[i]%k==0)
      //se la condizione dell'if è vera, a[i] si ricopia k volte
               {for (int j=0; j<k; j++)
                     {result[l]=a[i]; l++; }
               }
               else {result[l]=a[i]; l++; }
      // altrimenti si ricopia una volta sola
            }
  return result;}
```

complessità DI ENTRAMBE LE SOLUZIONI **θ(n*k) dove n**=a.length

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 numFigurineComuni (boolean[] album1, boolean[] album2)

  restituisce il numero di figurine che sono presenti in entrambi gli album. Ad esempio, dati in input:

  album1 = {false, false, true, true, false, true, false, true};

  album2 = {false, true, true, false, false, true, false, true};

  il metodo restituisce 3, in quanto le figurine di indice 2,5,7 sono presenti in entrambi gli album.

- int[] figurinePresenti (boolean[] album)

  restituisce un array contenente gli indici di tutte le figurine presenti nell'album. Ad esempio, dato in input l'album:

  album = {false, false, true, true, false, true, false, true};

  restituisce l'array {2,3,5,7}.