Esercizi riassuntivi
(Fondamenti di Informatica 2 – Walter Didimo)

Esercizio 1 Dire quale è la complessità temporale del seguente metodo, espressa con notazione asintotica O(.) (con la migliore approssimazione possibile) rispetto alle dimensioni della matrice a. Si motivi brevemente la risposta.

```
public static int esercizio (int[][] a){
  int m = a.length;
  int n = a[0].length;
  int i, somma=0;
  if (m <= n)
     i=m;
  else
     i=n;
  for (j=0; j<i; j++)
     somma += a[j][j];
  return somma;
}
```

Esercizio 2 Esprimere la complessità computazionale dei due metodi seguenti, usando la notazione asintotica O(.) (con la migliore approssimazione possibile) rispetto al numero n passato come parametro. Si motivi brevemente la risposta.

```
public double metodo1 (int n){
      double x = 0;
      for (int j=1; j<n; j++){
            int h = 0;
            while (h < j){
                  x = x + h;
                  h++;
            }
      }
      return x;
}


public double metodo2 (int n){
      double x = n;
      for (int j=1; j<3500; j++){
            int h = 0;
            while (h < j){
                  x = x + h;
                  h++;
            }
      }
      return x;
}
```

Esercizio 3 Scrivere un metodo di classe ricorsivo che riceve in ingresso una stringa e calcola il numero di vocali contenute in tale stringa (una soluzione non ricorsiva è considerata errata).

Esercizio 4 Scrivere un metodo di classe ricorsivo che riceve in ingresso una stringa ed un carattere e restituisce **true** se il carattere compare nella stringa e **false** altrimenti (una soluzione non ricorsiva è considerata errata).

Esercizio 5 La classe Arrays è una classe dell'API di Java già definita e pronta per l'uso. La classe Arrays si trova nel package java.util, ed ha un metodo statico (cioè di classe), avente il seguente prototipo: *public static void sort (int[ ] seq)*

L'invocazione *Arrays.sort (seq)* eseguirà l'ordinamento dei numeri nell'array seq.

Utilizzando il metodo *Arrays.sort (seq)*, completare il codice della seguente classe:

```
import java.util.*;

class Sequenza{

     /* ordina la sequenza seq e restituisce la massima differenza
     tra due elementi consecutivi */
     public static int maxDelta (int[] seq){
          ...
     }

     /* verifica il corretto funzionamento del metodo maxDelta */
     public static void main (String[] args){
          InputWindow in = new InputWindow ();
          int dim = in.readInt("NUmero elementi?");
          int[] seq = new int[dim];
          for (int i=0; i<seq.length; i++)
               seq[i] = in.readInt();
          int max = Sequenza.maxDelta(seq);
          System.out.println ("Max delta = " + max);
     }

}
```

Esercizio 6 Un oggetto della classe Insieme, permette di rappresentare un insieme di numeri di dimensione arbitraria. Un oggetto della classe Insieme, rappresentante inizialmente un insieme vuoto, può essere istanziato con questo costruttore:

- `public Insieme ( )`

Inoltre, su un oggetto Insieme, possono essere invocati questi due metodi:

- `public void add (int k);  // aggiunge l'intero k all'insieme`
- `public int removeMin ();  // rimuove un elemento di valore minimo`
  `// dall'insieme e lo restuisce.`

Supponendo che la classe Insieme sia già stata implementata, si chiede di spiegare cosa fa il seguente metodo di classe.

```
public static int[] met (int [] a){
      int[] s = new int[a.length];
      Insieme ins = new Insieme ();
      for (int i=0; i<a.length; i++)
            ins.add (a[i]);
      for (int i=0; i<a.length; i++)
            s[i] = ins.removeMin();
      return s;
}
```

Supponendo inoltre che, su un insieme di *m* elementi, una singola esecuzione del metodo *add* o del metodo *removeMin* richieda tempo O(log *m*), dire quale è la complessità asintotica del metodo *met* in funzione della dimensione *n* dell'array *a*.

Esercizio 7 Considera il seguente array di numeri interi:

10	**2**	**5**	**14**	**3**	**7**

Mostrare come l'array si modifica nelle prime tre passate dei seguenti algoritmi di ordinamento: selection sort, insertion sort, bubble sort.

Selection sort

1 passata						
2 passata						
3 passata						

Insertion sort

1 passata						
2 passata						
3 passata						

Bubble sort

1 passata						
2 passata						
3 passata						

Esercizio 8 Con riferimento al modello run time della Java Virtual Machine, dire quali delle seguenti affermazioni è vera o falsa, cerchiando la lettera V oppure la lettera F.

Una variabile di istanza corrisponde ad una locazione di memoria allocata nello heap	V	F
Una variabile di istanza corrisponde ad una locazione di memoria allocata nello stack	V	F
Un oggetto istanziato si trova nello stack	V	F
Il punto di ritorno di un metodo è un indirizzo nello heap	V	F
Ogni volta che viene invocato un metodo si alloca un record di attivazione	V	F
La deallocazione della memoria è un processo automatico eseguito dalla JVM	V	F

Esercizio 9 Dire cosa visualizza il seguente programma, mostrando lo sviluppo del calcolo:

```
class Esame{
      public static void main (String[]args){
            int x = 5;
            int y = f(x);
            System.out.println(y);
      }
      public static int f(int n){
            int z;
            if (n <=1) z = 1;
            else       z = 3 * f(n/2) + 4 * f (n%2);
            return z;
      }
}
```

Esercizio 10 Scrivere un metodo ricorsivo che calcola il resto della divisione intera tra un numero intero non negativo **i** e un numero intero positivo **j** (una soluzione non ricorsiva è considerata errata).