

Problema 1

Si considerino i seguenti metodi:

```
private static double mistero(double[] a, int i, double x) {
    if(i == a.length-1) return a[i];
    return a[i] + x * mistero(a, i+1, x);
}

public static double mistero(double[] a, double x) {
    return mistero(a, 0, x);
}
```

- Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `mistero(double[], double)` in funzione della dimensione dell'input.
- Riscrivere il metodo `mistero(double[], double)` senza utilizzare la ricorsione, mantenendone lo stesso costo computazionale.
- Spiegare come l'algoritmo `mistero` possa essere impiegato nella costruzione di una funzione hash.

Problema 2

- Definire il tipo astratto *coda di priorità* e descrivere lo schema di ordinamento su di esso basato, denominato PQ-sort.
- Istanziare (in Java) il PQ-sort nel caso di coda di priorità implementata con lista ordinata e confrontarlo con l'insertion-sort (descritto in Java).
- Descrivere lo heap-sort (Java o pseudo-codice) e confrontarlo con il PQ-sort basato su lista non ordinata (descritto in Java).

Problema 3

Un automa a stati finiti deterministico (ASF) può essere rappresentato tramite un multi-grafo orientato (sono permessi archi multipli e autoanelli), i cui nodi sono associati agli stati e i cui archi sono etichettati con i simboli dell'alfabeto dell'ASF.

- Realizzare la classe `ASF` (e le eventuali classi relative ad archi e nodi) che rappresenta un ASF. Le classi dovranno contenere, oltre alle variabili di istanza, anche le firme di tutti i metodi pubblici ritenuti fondamentali per risolvere i problemi successivi.
- Aggiungere alla classe `ASF` il metodo Java `int pruneUnreachable()` che modifichi `this ASF` eliminando gli stati non raggiungibili dallo stato iniziale e restituisca il numero di stati eliminati.
- Aggiungere alla classe `ASF` il metodo Java `boolean compute(String x)` che restituisce `true` se e solo se `this ASF` accetta la stringa `x`.