

Esercizio 1

Si consideri una classe *AlberoBinario* che rappresenta alberi binari che rappresentano espressioni. Si assuma che in tale classe siano implementati i seguenti metodi:

```
public interface AlberoBinario{
    /* restituisce il sottoalbero destro dell'albero corrente, la complessità temporale è  $\Theta(1)$  */
    public AlberoBinario destro();

    /* restituisce il sottoalbero sinistro dell'albero corrente, la complessità temporale è  $\Theta(1)$  */
    public AlberoBinario sinistro();

    /* Il metodo restituisce il valore memorizzato nella radice dell'albero. Se l'albero è una foglia il valore ha un significato altrimenti va ignorato. La complessità temporale è  $\Theta(1)$  */
    public int val();
}
```

Si deve realizzare un metodo

```
public static int countEqualTo(AlberoBinario a, int x) {...}
```

che calcola la numero dei nodi interni di *a* (compresa la radice se non è una foglia) aventi valore *x* che hanno entrambi i figli.

Si caratterizzi la complessità temporale e spaziale del metodo nel caso migliore e peggiore, specificando anche quali siano il caso migliore ed il caso peggiore per la complessità temporale e spaziale.

Caso Migliore:

1. Complessità temporale: $\Theta(\quad)$
2. Complessità spaziale: $\Theta(\quad)$

Caso Peggiore:

1. Complessità temporale: $\Theta(\quad)$
2. Complessità spaziale: $\Theta(\quad)$

Commenti: _____

Esercizio 2

Dire quali delle seguenti affermazioni sono vere e quali false.

	V	F	Affermazione
1			La complessità intrinseca del problema di ricercare un elemento in un array di interi è $\Omega(n)$, dove n è il numero di interi presenti nell'array.
2			L'algoritmo QuickSort ha complessità temporale $\theta(n * \lg n)$ nel caso peggiore.
3			La funzione $f(n) = \lg n^2$ è $O(n \lg n)$.
4			Sia G un grafo non orientato ed aciclico. Il grafo G è un albero.
5			Un albero binario è bilanciato se la differenza fra l'altezza del sottoalbero sinistro della radice e l'altezza del sottoalbero destro della radice è minore o uguale ad 1.
6			La complessità della ricerca di un elemento in una tabella hash in cui sono presenti n elementi nel caso migliore è $O(n)$.
7			La complessità spaziale della visita infiga di un albero è $\Omega(n^2)$, dove n è il numero di nodi presenti nell'albero.
8			La complessità dell'algoritmo di Floyd è $O(n^2)$, dove n è il numero di nodi nel grafo in input.
9			Dato un grafo orientato G se G contiene un ciclo allora c'è almeno un nodo con grado di entrata 0
10			In un grafo orientato un ciclo è un cammino che parte e finisce nello stesso nodo.

Esercizio 3

Si assuma di avere un algoritmo Divide et Impera che divide l'istanza problema originario (di dimensione n) in 4 istanze la cui dimensione è $n/2$. La fase di suddivisione del problema in sottoproblemi e di costruzione della soluzione dell'istanza originaria a partire dalle soluzioni delle istanze in cui essa è decomposta costa $b \cdot n^2$. Calcolare la complessità dell'algoritmo.
