

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 Θ(1) */  
    public AlberoBinario destro();  
  
    /* restituisce il sottoalbero sinistro dell'albero corrente, la complessità temporale è Θ(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 Θ(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 un solo figlio.

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: $\Theta(\underline{\hspace{2cm}})$
2. Complessità spaziale: $\Theta(\underline{\hspace{2cm}})$

Caso Peggio:

1. Complessità temporale: $\Theta(\underline{\hspace{2cm}})$
2. Complessità spaziale: $\Theta(\underline{\hspace{2cm}})$

Commenti:

Esercizio 2

Dire quali delle seguenti affermazioni sono vere e quali false.

	<i>V</i>	<i>F</i>	<i>Affermazione</i>
1			La complessità intrinseca del problema della ricerca di un elemento in un array ordinato di n elementi è $\Omega(n)$
2			Un grafo connesso contenente un ciclo di m archi ammette almeno m alberi ricoprenti.
3			In un grafo orientato non contenente cicli, esiste almeno un nodo con grado di entrata uguale a 0.
4			La complessità di accedere all'elemento con priorità massima in una coda di priorità realizzata tramite heap è $\Theta(\log n)$, essendo n il numero di elementi nella coda
5			Sia G un grafo non orientato e ciclico composto da n nodi. Il numero di archi di G è maggiore di o uguale a n .
6			L'algoritmo di <i>Prim</i> , preso in input un grafo pesato G , restituisce un albero ricoprente A tale che, per ogni coppia di nodi u, v , il cammino in A tra u e v è il cammino minimo tra u e v in G .
7			Il problema della ricerca del minimo elemento in un array ordinato ha complessità intrinseca $\Omega(n)$, dove n è il numero di elementi nell'array.
8			La complessità dell'algoritmo di <i>Floyd</i> è $O(n^3)$, dove n è il numero di nodi nel grafo in input.
9			Nel caso peggiore, ricercare un elemento in un albero AVL con n nodi ha complessità $O(\log n)$
10			Sia dato un algoritmo A risolutore del problema P . Se la complessità di A è $\Omega(f(n))$ allora la complessità intrinseca di P è $\Omega(f(n))$.

Exercise 3

Si assuma di avere un algoritmo Divide et Impera che divide l'istanza problema originario (di dimensione n) in 3 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 $b \cdot n$. Calcolare la complessità dell'algoritmo.