

Cognome: Nome: Matricola:

UNIVERSITÀ DEGLI STUDI DELLA CALABRIA
Corso di Laurea in Ingegneria Informatica

Prova scritta di *Algoritmi e Strutture Dati*
(durata della prova: 60 minuti)

Esercizio 1

Si consideri una classe *AlberoBinario* che rappresenta *alberi binari* in cui la parte informativa di ogni nodo è un numero intero. 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( );

    /* restituisce il valore memorizzato nella radice dell'albero, la complessità temporale è  $\theta(1)$ */
    public int val( );
}
```

Si deve realizzare un metodo ricorsivo

```
public static boolean verifica(AlberoBinario a) {...}
```

che restituisce *true* se e solo se almeno una delle foglie più profonde dell'albero contiene un valore maggiore o uguale a zero.

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: $\theta(\text{_____})$
2. Complessità spaziale: $\theta(\text{_____})$

Caso Peggior:

1. Complessità temporale: $\theta(\text{_____})$
2. Complessità spaziale: $\theta(\text{_____})$

[illegible]

Esercizio 2

Dire quali delle seguenti affermazioni sono vere e quali false.

	<i>V</i>	<i>F</i>	<i>Affermazione</i>
1			Può esistere un algoritmo che abbia complessità spaziale $\theta(n^2)$ e complessità temporale $\theta(n)$, dove n indica la dimensione dell'input.
2			La funzione $f(n) = n^2$ è $\Omega(n * \lg n)$.
3			L'algoritmo QuickSort ha complessità temporale $\theta(n * \lg n)$ nel caso peggiore.
4			L'inserimento di un elemento in un heap binario ha complessità temporale $\theta(\lg n)$ nel caso peggiore (dove n è il numero dei nodi).
5			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 $\Omega(f(n))$.
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			Sia G un grafo non orientato ed aciclico. Il grafo G è un albero.
8			Un grafo non orientato è connesso se e solo se ogni nodo è raggiungibile da ogni altro tramite un cammino.
9			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.
10			Un albero binario è detto di ricerca se, per ognuno dei suoi nodi u , la radice del figlio sinistro di u contiene un valore minore di quello contenuto in u e la radice del figlio destro di u contiene un valore maggiore o uguale di quello contenuto in u .

Esercizio 3

Preso un generico algoritmo divide et impera che ad ogni passo suddivide l'istanza corrente in a sotto-istanze dello stesso problema, ciascuna di dimensione n/c (dove n è la dimensione dell'istanza in esame e c una costante), si ricavi la complessità dell'algoritmo supponendo che al netto delle chiamate ricorsive la singola chiamata abbia complessità $b * n^d$, dove b e d sono costanti.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.