Cognome: | Nome: | Matricola:

UNIVERSITÀ DEGLI STUDI DELLA CALABRIA

Corso di Laurea in Ingegneria Informatica

Prova scritta di *Algoritmi e Strutture Dati*
TRACCIA A
(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 θ(1)*/
    public AlberoBinario destro( );
    /* restituisce il sottoalbero sinistro dell'albero corrente, la complessità temporale è θ(1)*/
    public AlberoBinario sinistro( );
    /* restituisce il valore memorizzato nella radice dell'albero, la complessità temporale è θ(1)*/
    public int val( );
}
```

Si deve realizzare un metodo ricorsivo

public static boolean verifica(AlberoBinario a) {…}

che restituisce *true* se e solo se esiste un nodo non foglia *n* di *a* tale che tutti i nodi foglia che appaiono nel sottoalbero radicato in *n* hanno lo stesso valore.

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: θ(____)
2. Complessità spaziale: θ(____)

Caso Peggiore:
1. Complessità temporale: θ(____)
2. Complessità spaziale: θ(____)

Commenti:_____

Esercizio 2

Dire quali delle seguenti affermazioni sono vere e quali false.

	V	*F*	*Affermazione*
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 *Prim*, 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 *Floyd* è $O(n^5)$, 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 $\Omega(f(n))$.

Esercizio 3

Preso un generico algoritmo divide et impera che ad ogni passo suddivide l'istanza corrente in *2* sotto-istanze dello stesso problema, ciascuna di dimensione *n/3* (dove *n* é la dimensione dell'istanza in esame), si ricavi la complessità dell'algoritmo supponendo che al netto delle chiamate ricorsive la singola chiamata abbia complessità *b*n*, dove *b* è una costante.