

Algoritmi e Strutture Dati

a.a. 2013/14

Prima prova intermedia del 23/01/2014

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

1. Dato un albero binario i cui nodi sono colorati di *bianco* o di *nero*, scrivere una **funzione C efficiente** che calcoli il numero di nodi aventi lo stesso numero di discendenti bianchi e neri. (Un nodo è discendente di se stesso.)
Inoltre analizzare la complessità di tale algoritmo.

Il tipo **Node** utilizzato per rappresentare l'albero binario è il seguente:

```
typedef struct node{  
    char * colore;  
    struct node * left;  
    struct node * right;  
} * Node;
```

2. Si può ordinare un dato insieme di n numeri costruendo un albero binario di ricerca che contiene questi numeri (usando ripetutamente *Tree-Insert* per inserire i numeri uno alla volta) e stampando poi i numeri utilizzando un certo tipo di visita. Scrivere l'algoritmo che realizza questo ordinamento e specificare il tipo di visita effettuata e il relativo algoritmo.

Quali sono i tempi di esecuzione nel caso peggiore e nel caso migliore per questo algoritmo di ordinamento?

3. Si definiscano formalmente le relazioni O , Ω , Θ e, **utilizzando le definizioni date e nient'altro**, si dimostri la verità o la falsità di ciascuna delle seguenti affermazioni:

a) $n \log n = \Theta(n^2)$

b) $n + \sqrt{3n} = \Omega(\log n)$

c) $4n \log n = O(4n + \log n^2)$

d) $2^{n+k} = O(2^n)$, dove k è una costante intera positiva

e) $2^{n+n} = O(2^n)$