

Algoritmi e Strutture Dati

a.a. 2014/15

Prima prova intermedia del 28/01/2015

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

1. Un nodo di un albero binario **u** è detto **intermedio** se la somma delle chiavi contenute nei nodi del sottoalbero di cui **u** è radice è uguale alla somma delle chiavi contenute nei nodi sul percorso che collega **u** alla radice dell'albero (**u** escluso).
 - a) Scrivere una funzione **efficiente** in C che restituisca il numero di nodi intermedi.
 - b) Analizzare la complessità della soluzione trovata, indicando la corrispondente relazione di ricorrenza.
 - c) Se vogliamo modificare la funzione in modo che restituisca l'**insieme** dei nodi intermedi che tipo di struttura dati si può utilizzare per rappresentare l'insieme? La complessità dell'algoritmo deve rimanere la stessa che nel caso a).

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

```
typedef struct node{  
    int key;  
    struct node * left;  
    struct node * right;  
} * Node;
```

2. Scrivere un algoritmo **efficiente**, di tipo **divide et impera**, che conta il numero di occorrenze della sequenza 'a' 'r' memorizzata in posizioni adiacenti in un array di caratteri.

Analizzare la complessità indicando e risolvendo la corrispondente relazione di ricorrenza.

Esempio: Per l'array <a, b, c, r> la risposta sarà 0.

Per l'array <b, a, r, c, a, r> la risposta sarà 2.

3. Si enunci e si dimostri il teorema fondamentale delle ricorrenze e lo si utilizzi per risolvere le seguenti ricorrenze (spiegando in quali casi del teorema ricade ciascuna di esse):

- $T(n) = 3T(n/2) + n^2$

- $T(n) = 4T(n/2) + n^2$

- $T(n) = T(n/2) + 2^n$