

Algoritmi e Strutture Dati

a.a. 2017/18

Prima prova intermedia del 16/01/2018

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

1. Scrivere una **funzione C efficiente**, di nome **simmetrico**, che, dato un albero binario, ritorna *1* se l'albero è **speculare**, sia dal punto di vista strutturale che nel contenuto dei nodi, altrimenti ritorna *0*.

Specificare la chiamata della funzione nel **main**.

Analizzare la complessità della funzione.

2. Sia BST^+ la struttura dati che si ottiene aggiungendo ad ogni nodo x di un albero binario di ricerca un nuovo attributo **diff** che contiene la differenza fra il numero di nodi nel sottoalbero sinistro e quelli nel sottoalbero destro di x .

Modificando la procedura **Tree-insert** si definisca una procedura **BST^+ -insert** per l'inserimento di una nuova chiave in un BST^+ .

Il prototipo della procedura è:

BST^+ -insert(TreeConDiff t, NodeConDiff z)

Si assuma che il nodo z sia così inizializzato:

$z.p = z.left = z.right = NULL$

$z.key = k$ (nuova chiave)

$z.diff = 0$

Analizzare la complessità della procedura.

3. Si stabilisca se le seguenti affermazioni sono vere o false:

- $n \log n = \Theta(n^2)$
- $4n \log n = O(4n + \log n^2)$
- $2^n = O(n!)$
- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, allora $f(n) = O(h(n))$
- Se $f(n) = O(g(n))$, allora $g(n) = \Omega(f(n))$

Si forniscano giustificazioni formali. In caso contrario l'esercizio non verrà valutato pienamente, anche in presenza di risposte corrette.