

# Algoritmi e Strutture Dati

a.a. 2018/19

## Prima prova intermedia del 25/01/2019

Cognome: \_\_\_\_\_

Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_

E-mail: \_\_\_\_\_

1. (a) Dato un albero binario  $T$ , definire l'altezza di un nodo  $v$ .  
(b) Sia  $T$  un albero binario i cui nodi  $x$  hanno campi *left*, *right* e *key*. Scrivere una **funzione in C efficiente**  $altezzaNodi(u, k)$  che riceve in input la radice  $u$  di un albero binario  $T$  e un numero naturale  $k$  e restituisce il numero di nodi che hanno altezza  $k$ .  
(c) Analizzare la complessità della funzione, indicando eventuali relazioni di ricorrenza.

**Non si possono utilizzare strutture dati ausiliarie**

2. Sia dato un albero binario di ricerca  $T$  con  $n$  nodi e chiavi naturali. Si scriva tramite pseudocodice una funzione **efficiente** che restituisca un albero binario di ricerca  $T'$  contenente tutte e sole le chiavi **pari** di  $T$ .  
Qual è la complessità in termini di tempo della funzione proposta? La soluzione proposta è in loco?

**Giustificare le risposte.**

3. Per un certo problema sono stati trovati tre algoritmi risolutivi ( $A_1$ ,  $A_2$  e  $A_3$ ) con i seguenti tempi di esecuzione (dove  $n$  rappresenta la dimensione dei dati di ingresso):

$$A_1: \quad T(n) = 4n/2 + \log n$$

$$A_2: \quad T(n) = 4T(n/2) + \log n$$

$$A_3: \quad T(n) = 4T(n/2) + 2^n$$

Si dica quale dei tre algoritmi è preferibile per  $n$  sufficientemente grande. Si forniscano giustificazioni formali. In caso contrario l'esercizio non verrà valutato pienamente, anche in presenza di risposte corrette.