

# Introduzione agli Algoritmi

## Esame Scritto a canali unificati

docenti: T. Calamoneri, A. Monti  
Sapienza Università di Roma  
17 Gennaio 2024

### Esercizio 1 (10 punti):

Si risolva con il **metodo dell'albero** la seguente equazione di ricorrenza:

$$\begin{cases} T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + \Theta(n) & \text{se } n > 1 \\ T(1) = \Theta(1) \end{cases}$$

Si dettagliano i ragionamenti ed i passaggi del calcolo, giustificando ogni affermazione.

### Esercizio 2 (10 punti):

Si scriva lo pseudocodice, opportunamente commentato, di una funzione **iterativa** che, preso in input un array  $A$  di interi non nulli (positivi e negativi), ne sposti gli elementi in modo che gli interi negativi precedano gli interi positivi e restituisca poi l'array così modificato.

Ad esempio per  $A = [3, -5, -7, 1, -8]$  due possibili risposte corrette (ma ve ne sono altre) sono  $A = [-8, -7, -5, 1, 3]$  o  $A = [-5, -8, -7, 3, 1]$ .

La funzione deve avere costo computazionale  $O(n)$ , dove  $n$  è il numero di elementi presenti nell'array. Il costo in termini di spazio oltre l'array  $A$  deve essere  $\Theta(1)$  (in pratica non può far uso di array di appoggio).

Si motivi in dettaglio il costo computazionale dell'algoritmo proposto.

Si risponda, inoltre, alle seguenti domande:

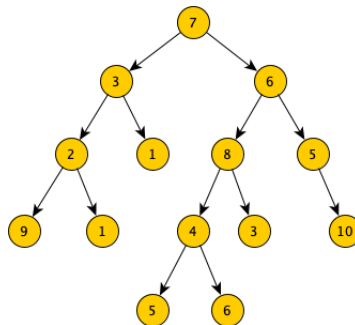
- qual è il costo computazionale se l'array in input è ordinato in modo non decrescente?
- qual è il costo computazionale se l'array in input è ordinato in modo non crescente?
- qual è il costo computazionale se l'array in input è costituito da elementi tutti uguali?

### Esercizio 3 (10 punti):

Si consideri un albero binario radicato  $T$ , i cui nodi abbiano un campo *val* contenente un intero e i campi *left* e *right* con i puntatori ai figli. Un nodo  $v$  dell'albero si dice *valido* se si verificano le seguenti tre condizioni:

- $v$  ha entrambi i figli;
- il campo *val* di uno dei due figli è minore di quello padre;
- il campo *val* dell'altro figlio è maggiore di quello del padre.

Ad esempio nell'albero in figura ci sono esattamente 2 nodi validi (quelli con valore 6 e 2).



Si progetti una **funzione ricorsiva** che, dato il puntatore  $r$  alla radice di  $T$  restituisca il numero di nodi validi di  $T$  in tempo  $\mathcal{O}(n)$  dove  $n$  è il numero di nodi presenti nell'albero.

Della funzione proposta:

- si dia la descrizione a parole,
- si scriva lo pseudocodice,
- si giustifichi formalmente il costo computazionale.

NOTA BENE: nello pseudocodice della funzione ricorsiva **non** si deve far uso di variabili globali.