Algoritmi e Strutture Dati

Esercizio 1.[10 punti]
Risolvere in ordine di grandezza (senza usare il master theorem) la seguente equazione ricorsiva:

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ T(\frac{12n}{27}) + T(\frac{30n}{54}) + n & \text{if } n > 1. \end{cases}$$

Traccia della soluzione dell'esercizio .
$T(n) = \Theta(n \log(n))$. Per dimostrare che $T(n) = \Theta(n \log(n))$ è possibile utilizzare l'albero della ricorsione. Per dimostrare che $T(n) = O(n \log(n))$, si noti che l'albero risulterà non bilanciato, ma la lunghezza del cammino radice-foglia più lungo sarà comunque logaritmica in n. Ogni livello dell'albero avrà un costo minore o uguale a n. Per dimostrare che $T(n) = \Omega(n \log(n))$, si noti che l'albero risulterà non bilanciato, ma la lunghezza del cammino radice-foglia più corto sarà comunque logaritmica in n.
L'esercizio poteva essere risolto anche dimostrando che $T(n) = \Theta(n \log(n))$ per induzione.

Esercizio 2.[11 punti]
Sia A un albero binario i cui nodi contengono un campo COLORE che può assumere il valore ROSSO, NERO o AZZURRO, un campo numerico KEY e due puntatori (uno al figlio sinistro e uno al figlio destro). Scrivere una funzione RICORSIVA che prenda in input A e restituisca in output il numero di nodi con esattamente 2 figli e con COLORE=NERO e KEY compreso tra 1 e 10. La funzione deve essere ricorsiva e scritta in pseudo codice. Analizzare il costo computazionale della funzione proposta.

Traccia Soluzione esercizio .
Input: puntatore p alla radice dell'albero
Caso base: se $p = nil$ ritornare 0
Se $p \neq nil$, si chiama la funzione ricorsivamente sul figlio di destra e di sinistra e si restituisce la somma del risultato delle due chiamate ricorsive, eventualmente aumentato di 1 nel caso in cui il nodo puntato da p soddisfi le condizioni indicate nel testo dell'esercizio.

Esercizio 3.[11 punti]
Sia $S = \{s_1, \ldots, s_n\}$ un insieme di segmenti. L'i-esimo segmento è rappresentato da un inizio e una fine:

$$s_i = (inizio, fine)$$

con $inizio \leq fine$. Scrivere una funzione che preso in input un insieme di segmenti S, calcoli il numero massimo di segmenti di S a due a due non sovrapposti. Due segmenti (a, b) e (c, d) sono sovrapposti quando esiste almeno un numero x tale che $a \leq x \leq b$ e contemporaneamente $c \leq x \leq d$. La funzione deve essere ricorsiva e scritta in pseudo codice. Analizzare il costo computazionale della procedura proposta e argomentarne la correttezza.

Traccia Soluzione esercizio .
Il problema è facilmente riconducibile al problema della "selezione di attività" visto in classe.