Prof. L. Margara **Scritto del 7/6/2021**

Algoritmi e Strutture Dati

Esercizio 1.[10 punti]
Risolvere in ordine di grandezza la seguente equazione ricorsiva.

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1, \\ T(n-1) + \frac{n^3}{\sqrt{n}} & \text{if } n > 1. \end{cases}$$

Soluzione esercizio .
Per errore ho messo fuori inizialmente la soluzione di $T(n) = T(n-1) + \sqrt{n^3}$ invece della soluzione di $T(n) = T(n-1) + \frac{n^3}{\sqrt{n}}$.
Quella che segue è la soluzione dell'equazione corretta.
Riscriviamo $T(n) = T(n-1) + n^{3-1/2} = T(n-1) + n^{5/2}$.
Svolgendo l'equazione otteniamo:

$$T(n) = n^{5/2} + (n-1)^{5/2} + (n-2)^{5/2} + \cdots + (\frac{n}{2})^{5/2} + \cdots + (3)^{5/2} + (2)^{5/2} + (1)^{5/2}$$

Questa quantitá puó essere maggiorata e minorata come segue:

$$T(n) \geq n^{5/2} + (n-1)^{5/2} + (n-2)^{5/2} + \cdots + (\frac{n}{2})^{5/2} \geq \frac{n}{2}(\frac{n}{2})^{5/2}$$

e quindi

$$T(n) \geq \frac{n}{2}(\frac{n}{2})^{5/2} = \Theta(n^{7/2}).$$

Inoltre banalmente abbiamo

$$T(n) \leq n\ (n)^{5/2} = \Theta(n^{7/2})$$

Quindi $T(n) = \Theta(n^{7/2})$.

Esercizio 2.[11 punti]
Scrivere una funzione F di costo lineare che prenda in input un vettore V (non ordinato) contenente n interi positivi e restituisca 1 se nel vettore V esistono almeno tre numeri x_1, x_2, x_3 distinti ripetuti almeno $\lceil n/10 \rceil$ volte, 0 altrimenti. Dimostrare la correttezza dell'algoritmo fornito e che il suo costo computazionale è $O(n)$.

Traccia Soluzione esercizio
Gli unici candidati c_i al ruolo di x_1, x_2, x_3 sono gli elementi calcolati da opportune "Select" (come visto a lezione). In questo caso, per stare tranquilli basta calcolare il risultato di 11 "Select":

$$c_i = Select[V, 1, n, i \cdot n/11], \quad i = 1, \ldots, 11$$

A questo punto basta controllare che esistano tre elementi distinti tra tutti i c_i ripetuti almeno $\lceil n/10 \rceil$ in V.

Esercizio 3.[12 punti]
Scrivere una procedura RICORSIVA che prenda in input un albero binario A e restituisca in output un albero B ottenuto da A eliminando tutti e soli i nodi che hanno esattamente un figlio. La procedura deve essere ricorsiva e scritta in pseudo codice. Analizzare il costo computazionale della procedura proposta.

Traccia Soluzione esercizio
La soluzione di questo esercizio è facilmente ricostruibile dalla soluzione di esercizi simili svolti a lezione e presenti nel materiale didattico.
L'idea è quella di visitare l'albero ricorsivamente e di controllare se il nodo corrente nella visita ha un figlio (sinistro o destro) che a sua volta ha un solo figlio. Facendo questo controllo sul padre dell'eventuale nodo da eliminare siamo in grado di collegare il nodo corrente direttamente con il nipote cancellando in questo modo il nodo che deve essere cancellato senza dover risalire di un livello nell'albero.

UnFiglioSolo(p)

```
1  if p == nil
2        return False
3  n = 0
4  if p.left == nil
5        n = n + 1
6  if p.right == nil
7        n = n + 1
8  if n == 1
9        return True
10 return False
```

Figlio(p)

```
1  if p.left == nil
2        return p.right
3  if p.right == nil
4        return p.left
```

```
CancellaNodi(p)
1  if p ≠ nil
2       if UnFiglioSolo(p.left)
3            p.left = Figlio(p.left)
4       if UnFiglioSolo(p.right)
5            p.right = Figlio(p.right)
6       CancellaNodi(p.right)
7       CancellaNodi(p.left)
```