

## Algoritmi e Strutture Dati

### Esercizio 1.[11 punti]

$$S(n) = \begin{cases} 1 & \text{if } n \leq 2, \\ T(n-1) + \log(n^{12}) & \text{if } n > 2. \end{cases}$$
$$T(n) = \begin{cases} 1 & \text{if } n \leq 2, \\ T(n-1) + \log(n^{24}) & \text{if } n > 2. \end{cases}$$

Dire se  $S(n) = o(T(n))$  oppure  $S(n) = \omega(T(n))$  oppure  $S(n) = \Theta(T(n))$ .  
Giustificare formalmente la risposta.

**Traccia della soluzione dell'esercizio .**

$$T(n) = \Theta(S(n))$$

L'esponente all'interno del logaritmo diventa una costante moltiplicativa se portata all'esterno.

$$T(n) = S(n) = n \log(n)$$

**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  $V$  esiste un numero  $x$  ripetuto almeno  $\lceil 0,4 \cdot n \rceil$  volte, 0 altrimenti. Dimostrare la correttezza dell'algoritmo fornito e che il suo costo computazionale è  $O(n)$ . (suggerimento: usare la Selection)

**Traccia Soluzione esercizio**

Si usano due Select chiamate su posizioni del vettore scelte opportunamente, ad esempio in posizioni  $n/3$  e  $2n/3$ . I due risultati delle Select sono gli unici candidati ad essere gli elementi richiesti dall'esercizio. A questo punto basta verificarli entrambi.

**Esercizio 3.[11 punti]**

Scrivere una procedura RICORSIVA che prenda in input un albero binario A e restituisca in output un albero B ottenuto da A eliminando tutte le foglie che hanno un valore uguale a quello del padre. La procedura deve essere ricorsiva e scritta in pseudo codice. Commentare il codice. Analizzare il costo computazionale della procedura proposta.

**Traccia Soluzione esercizio .**

Esercizio svolto in classe e presente nel materiale didattico.