

Algoritmi e Strutture Dati

Esercizio 1.[12 punti]

Scrivere una funzione ricorsiva F_1 che prenda in input un numero intero positivo n e restituisca in output 0. F_1 deve avere costo computazionale $\Theta(n^2)$. Analizzare il costo computazionale della funzione proposta.

Scrivere una funzione ricorsiva F_2 che prenda in input un numero intero positivo n e restituisca in output 0. F_2 deve avere costo computazionale $\Theta(\log(\log(n)))$. Analizzare il costo computazionale della funzione proposta.

Scrivere una funzione ricorsiva F_3 che prenda in input un numero intero positivo n e restituisca in output 0. F_3 deve avere costo computazionale $\Theta(5^n)$. Analizzare il costo computazionale della funzione proposta.

Utilizzare SOLAMENTE l'istruzione IF THEN ELSE, l'assegnamento e la chiamata ricorsiva (oltre che ovviamente il RETURN). Vietato utilizzare cicli (for, repeat, while, ...) o salti (goto, ...). Utilizzare la ricorsione.

Esercizio 2.[9 punti]

Calcolare la formula chiusa esatta per $T(n)$ definita come segue:

$$T(n) = \begin{cases} 77 & \text{if } n = 1, \\ T(n-1) + (-1)^n & \text{if } n > 1. \end{cases}$$

Esercizio 3.[12 punti]

Bip-Dominating-Set: Dato un grafo bipartito (L, R, E) e un intero positivo k dire se esiste un sottoinsieme di nodi di L di cardinalità minore o uguale a k che domini tutti i nodi in R (un insieme di nodi A domina un insieme di nodi B se ogni nodo di B è raggiunto da almeno un arco che parte da un nodo di A).

Trip-Dominating-Set: Dato un grafo tripartito (L, C, R, E) e un intero positivo k dire se esiste un sottoinsieme di nodi di L di cardinalità minore o uguale a k che domini contemporaneamente tutti i nodi in R e in C .

Dimostrare che se Trip-Dominating-Set si risolve in tempo polinomiale allora anche Bip-Dominating-Set si risolve in tempo polinomiale.