1 Esercizi

Esercizio 1 - Esame del 25-03-2022

Si risolva la seguente equazione di ricorrenza calcolandone l' **andamento asintotico**

$$T(n) = \begin{cases} 1 & \text{se } n \leq 2 \\ 3 \cdot T\left(\sqrt{n}\right) + 2 \cdot T \sqrt[4]{n} + \log n & \text{altrimenti} \end{cases}$$

Esercizio 2 - Esame del 25-03-2022

Si scriva un algoritmo che, dati un grafo $G = \langle V, E, \rangle$ e due vertici `s, u` $\in V$, verifichi in **tempo lineare** sulla dimensione del grafo se **tutti i percorsi infiniti** che **partono** da `s` **non passano infinite volte** per `u`

Esercizio 1 - Esame del 21-06-2022

Si individuino, nel caso esistano le **costanti moltiplicative** atte a mostrare la seguente relazione asintotica

$$\ln\left(\frac{n}{e}\right) = \Theta\left(\ln\left(n^e\right)\right)$$

(Si ricorda che con "ln" si indica il logaritmo naturale e con "e" la costante di Eulero, detta anche di Nepero)

In caso contrario mostrare la falsità della relazione

Esercizio 2 - Esame del 21-06-2022

Si scriva un **algoritmo ricorsivo** che, dati in ingresso un albero binario di ricerca su interi $\mathcal{T}$ e due valori $k_1, k_2 \in \mathbb{N}$, inserisca in una lista $\mathcal{L}$ le chiavi `k` contenute in $\mathcal{T}$ comprese tra k_1 e k_2 ($k_1 \leq k \leq k_2$), in modo che al termine $\mathcal{L}$ contenga valori ordinati in modo decrescente.

Tale algoritmo dovrà avere **complessità lineare** nella dimensione dell'albero.

Infine si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo di cui sopra

Esercizio 1 - Esame del 21-07-2022

Si individuino nel caso esistano, le **costanti moltiplicative** atte a mostrare la seguente relazione asintotica

$$\log_2(n^{2n}) - \log_2(n) = \Theta(\log_2(n^n))$$

In caso contrario mostrare la falsità della relazione

Esercizio 2 - Esame del 21-07-2022

Si scriva un **algoritmo ricorsivo** che, dati in ingresso un albero binario di ricerca su interi $\mathcal{T}$ e due valori $k_1, k_2 \in \mathbb{N}$, cancelli da $\mathcal{T}$ le chiavi k comprese tra k_1 e k_2 $(k_1 \leq k \leq k_2)$.

Tale algoritmo dovrà essere efficiente e non far uso **nè di variabili globali nè di parametri passai per riferimento**.

Infine si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo ricorsivo di cui sopra.

Esercizio 1 - Esame del 22-09-2022

Si individuino, nel caso esistano le **costanti moltiplicative** atte a mostrare la seguente relazione asintotica

$$n \cdot \ln(e^2 \cdot n) = \Theta(\ln \sqrt{(n)^n})$$

(Si ricorda che con "ln" si indica il logaritmo naturale e con "e" la costante di Eulero, detta anche di Nepero)

In caso contrario mostrare la falsità della relazione. Il procedimento seguito va riportato per esteso

Esercizio 2 - Esame del 22-09-2022

Si scriva un algoritmo che, dato in ingresso un grafo orientato $G = \langle V, E, \rangle$, verifichi **in tempo lineare sulla dimensione del grafo** se la seguente condizione è soddisfatta:

- G contiene 3 vertici distinti, di segutio denominati `a, b, c`, tali che:
  - `a` e `b` son entrambi raggiungibili da `c`;
  - `a` e `c` son entrambi raggiungibili da `b`;
  - `b` e `c` son entrambi raggiungibili da `a`;

Notare che i vertici `a, b, c` **non sono input** del problema

Esercizio 1 - Esame del 25-01-2023

Si risolva la seguente equazione di ricorrenza, calcolandone l' **andamento asintotico**

$$T(n) = \begin{cases} 1 & \text{se n} \leq 2 \\ 2 \cdot T \sqrt[4]{n} + \log(2n) & \text{altrimenti} \end{cases}$$

Esercizio 2 - Esame del 25-01-2023

Si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo ricorsivo di seguito riportato, dove Z è una funzione esterna non meglio specificata

```
function Algoritmo(T, h)
if T = Nil then
    return Z(0, h)
else
    a ← 0
    if T → key = 0 mod 2 then
        a ← a + Algoritmo(T → dx, 2 * h)

    if T → key = 1 mod 3 then
        a ← a + Algoritmo(T  sx, 3 * h)
    return Z(T -> key, a)
```

Esercizio 1 - Esame del 22-02-2023

Si risolva la seguente equazione di ricorrenza, calcolandone l' **andamento asintotico**

$$T(n) = \begin{cases} 1 & \text{se n} \leq 27 \\ 3\, n?2 \cdot T \sqrt[3]{n} + 2\, n^3 & \text{altrimenti} \end{cases}$$

Esercizio 2 - Esame del 25-01-2023

Si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo ricorsivo di seguito riportato, dove Z_t e Z_r sono 2 funzion esterne non meglio specificate che soddisfano la seguente proprietà

$$p < Z_t(A, p, s) < Z_r(A, p\, s) \leq s$$

quando $p + 1 < s$

```
function Algoritmo(A, p, s)
if s ≤ p + 1 then
    return 0
else
```

q ← $Z_t(A, p, s)$
r ← $Z_r(A, p, s)$
a ← $Algoritmo(A, p, q)$
a ← a − $Algoritmo(A, q, s)$
a ← a + $Algoritmo(A, r, s)$
return a + (r - q)