

Cognome:..... Nome:..... Matr.:.....

Esercizio 1 [16 punti]

A: *notazione asintotica*. Dire quali delle seguenti relazioni asintotiche sono vere:

$$\begin{aligned} n + n^2 \log^2 n &= o(n^2 \log n); & \log^4 n &= o(\sqrt[3]{n}); & n^2 &= \Omega\left(\frac{n^2}{\log^{2001} n}\right); & \frac{n\sqrt{n+\log n}}{\sqrt{n^3+3}} &= \Theta(\log n); \\ 2^{2n} &= \omega(2^{1.9n}); & 2^n &= \Theta(2^n + 1.5^n); & 2^n &= o(2^n + n^2); & 2^n &= \Theta(2^{n+8}); \end{aligned}$$

B: *equazioni di ricorrenza*. Fornire la soluzione asintotica alle seguenti relazioni di ricorrenza:

$$T(n) = T\left(\frac{99}{100}n\right) + n; \quad \text{Soluzione:}$$

$$T(n) = T(n-1) + n^3; \quad \text{Soluzione:}$$

C: *algoritmi e complessità*. Quale algoritmo useresti e quanto costa se devi:

- In un grafo diretto dire se esiste un nodo t che non può essere raggiunto da almeno un nodo s :
- In un grafo non orientato e pesato, individuare il cammino più corto da s a t che non passa per uno specifico nodo w :
- Costruire un albero AVL contenente n chiavi fornite in input:
- Fondere due heap binari, uno contenente n^2 nodi e l'altro n nodi:

Esercizio 2 [8 punti]

Sia $A[1 : n]$ un vettore di n numeri. Un k -picco in A è un indice $m \in \{k+1, \dots, n-k\}$ tale che la sequenza di elementi $A[m-k; m]$ è strettamente crescente, mentre la sequenza di elementi $A[m; m+k]$ è strettamente decrescente.

Si progetti un algoritmo che dato A calcola il più grande valore di k per cui A contiene un k -picco. L'algoritmo deve avere complessità $O(n)$. Si fornisca lo pseudocodice dettagliato.

Esercizio 3 [8 punti]

Nell'ultimo gioco rilasciato dalla *Mintendo*, Super Ciano Bross si trova su un nodo s di un grafo orientato $G = (V, E)$ con n nodi ed m archi, e deve raggiungere il nodo t per vincere il livello. Ogni arco e è associato inizialmente uno stato $\sigma(e) \in \{\text{on}, \text{off}\}$. Super Ciano può attraversare solo gli archi che sono nello stato **on**. C'è inoltre un insieme di nodi $B \subseteq V$ che contengono un bottone speciale. Se Ciano è su un nodo $b \in B$ può decidere di schiacciare il bottone e tutti gli archi invertono il proprio stato, quelli che erano nello stato **on** passano allo stato **off** e quelli che erano nello stato **off** passano nello stato **on**.

Progettate un algoritmo di complessità $O(m+n)$ che calcola, se esiste, una strategia per Super Ciano che lo porta a vincere il livello nel nodo t .