

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

Esercizio 1 [16 punti]

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

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

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

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

$$T(n) = 2T(n-4) + 1; \quad \text{Soluzione:}$$

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

- Ordinare n studenti laureati in informatica rispetto al voto ottenuto all'esame di Algoritmi e Strutture Dati:
- Restituire la lista dei nodi di un BST in ordine decrescente di chiave:
- Capire se tutti i nodi di un grafo diretto sono mutuamente raggiungibili:
- Capire se in un grafo diretto pesato, tutti i nodi sono raggiungibili a partire da un nodo sorgente s usando solo archi di peso al più x :

Esercizio 2 [6 punti]

Sia T un albero binario con n , dove ogni nodo v di T ha un colore $v.col$ che può essere *Blu* (B) o *Giallo* (G), e quindi $v.col \in \{B, G\}$. Si progetti un algoritmo che dato T e due interni b e g , restituisca il numero di nodi di T che hanno almeno b antenati blue o almeno g antenati gialli.

Si assuma che T è rappresentato tramite una struttura dati collegata, con record e puntatori, dove il record di ogni nodo, oltre al campo $v.col$, contiene anche il puntatore al figlio sinistro e al figlio destro del nodo. L'algoritmo deve avere complessità $O(n)$. Si fornisca lo pseudocodice dettagliato.

Esercizio 3 [10 punti]

Ti verranno forniti n numeri a_1, \dots, a_n e il tuo obiettivo è progettare un algoritmo che selezioni i k numeri più piccoli della collezione, dove k è un parametro intero che soddisfa $k = \Theta(\sqrt{n})$.

I numeri non vengono passati all'algoritmo attraverso un vettore, ma forniti in modalità *online*, ovvero snocciolati uno dopo l'altro, in sequenza. I numeri sono davvero molti ed è impensabile memorizzarli tutti. In particolare, è richiesto che l'algoritmo usi memoria $O(k)$. Come già detto l'algoritmo deve restituire alla fine i k valori più piccoli della collezione.

Progettate quindi un algoritmo efficiente che risolve il problema usando memoria $O(k)$. Il punteggio pieno può essere ottenuto solo se l'algoritmo ha complessità temporale $o(n\sqrt{n})$, ma anche soluzioni temporalmente meno efficienti verranno prese in considerazione.