

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

Esercizio 1 [16 punti]

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

$$n \log n + \sqrt{n} = O(n \log n); \quad \sqrt{\log n} = \omega(\log \log n); \quad n\sqrt{n} = \Theta(\frac{n^2+1}{\sqrt{n}}); \quad 2^{n+20} = \Theta(2^n);$$

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

$$T(n) = 2T(n/8) + 1; \quad \text{Soluzione:}$$

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

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

- Costruire un heap binario di n elementi;
- Trovare il secondo massimo in un vettore di n numeri;
- Dato un grafo diretto, capire se esistono due nodi s e t tale che s può raggiungere t ma non viceversa;
- Dato un grafo diretto G con pesi positivi sugli archi, un nodo s e un valore L , dire se esiste un nodo t la cui distanza verso s è più grande di L :

Esercizio 2 [8 punti]

Il dottor maestro Straziota sta facendo la collezione di figurine dei più grandi musicisti della storia. L'album completo è composto da N musicisti, che possiamo pensare numerati da 1 a N . Lui per il momento ne ha un piccolo sottoinsieme A di dimensione n , con $n << N$, e non gli dispiacerebbe un giorno completare l'album.¹ Le figurine escono dentro dei pacchetti di patatine. Ogni pacchetto contiene 3 figurine, ma ovviamente non è possibile vedere quali sono senza aprire la confezione. Così Straziota compra k pacchetti di patatine, P_1, \dots, P_k , con $P_i \subseteq \{1, 2, \dots, N\}$ e torna a casa per aprirli.

Progettate un algoritmo efficiente che presi in input A, P_1, \dots, P_k calcola il numero di figurine che ancora mancano dopo quest'ultimo acquisto. L'algoritmo deve avere complessità temporale $o(nk)$ e deve usare $o(N)$ memoria ausiliaria.

Esercizio 3 [8 punti]

La rete di trasporti pubblici della vostra città è modellata come un grafo non orientato $G = (V, E)$ dove ogni nodo v rappresenta una stazione e ogni arco (u, v) una tratta diretta fra le stazioni u e v . Non tutte le tratte sono servite dalla stessa compagnia e possono essere percorse con lo stesso biglietto. In particolare, ci sono k diversi livelli di abbonamento, numerati da 1 a k , di costo crescente, ed ogni arco e è etichettato con un intero $\lambda(e)$, che specifica che quella tratta è percorribile se si ha a disposizione un abbonamento di livello $\lambda(e)$ o superiore.

¹Pensate ad N come ad un numero molto più grande di n , per esempio $N = n^3$.

Voi abitate vicino alla stazione s e volete comprare un abbonamento che vi garantisca di poter raggiungere ogni altra stazione della città in al più h tratte. Ovviamente volete spendere il meno possibile. Progettate un algoritmo che calcola il più economico abbonamento che rispetta il vincolo di qualità di servizio. Per ottenere il punteggio pieno la complessità dell'algoritmo deve essere $O(m \log m)$, ma anche soluzioni meno efficienti saranno valutate.