

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

Esercizio 1 [16 punti]

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

$$n \log n + n^2 = O(n \log n); \quad \log^2 n = o(\sqrt{n}); \quad n\sqrt{n} = \Theta\left(\frac{n^2+1}{\sqrt{n}}\right); \quad 2^{1.5n} = \omega(2^n);$$

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

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

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

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

- Permutare n numeri in un vettore rispetto ad un elemento perno, in modo che compaiano prima gli elementi più piccoli o uguali del perno, poi il perno e poi quelli più grandi:
- Fare il *merge* di due vettori ordinati in uno ordinato:
- Ordinare n interi compresi fra n e n^2 :
- Dato un grafo diretto G , dire se tutti i nodi possono raggiungere uno specifico nodo t attraversando al più h archi:

Esercizio 2 [8 punti]

Sia T un albero binario radicato di n nodi in cui ogni nodo v può essere *bianco* (B) o *nero* (N). Diciamo che un nodo è *grigio* se il numero dei suoi antenati neri è uguale a quello dei suoi antenati bianchi. Progettare un algoritmo di complessità $O(n)$ che restituisce il numero di nodi grigi di T . Si assuma che T è mantenuto con una struttura dati collegata, fatta di record e puntatori, e che il record di un nodo v contenga i seguenti campi: il puntatore al figlio sinistro $sin(v)$, quello al figlio destro $des(v)$, e un campo $col(v) \in \{B, N\}$. Si fornisca lo pseudocodice dettagliato dell'algoritmo.

Esercizio 3 [8 punti]

Un labirinto incantato è modellato come un grafo diretto $G = (V, E)$. Tu ti trovi sul nodo s e l'uscita è sul nodo t . Hai inizialmente una riserva di forza di Δ unità e conosci, per ogni arco e del grafo, il numero di unità di forza $c(e)$ che devi spendere per attraversare l'arco. Nel labirinto ci sono anche delle stanze speciali corrispondenti a un sottoinsieme U di nodi del grafo in cui puoi effettuare una speciale magia. In particolare, per ogni coppia di nodi x e y di U , puoi scomparire da x e ricomparire in y . Questo però ti costa $\gamma(x, y)$ unità di forza se esegui questo tipo di magia la prima volta, mentre $5\gamma(x, y)$ unità di forza la seconda volta.¹ Ti disintegreresti immediatamente se provassi a scomparire una terza volta. Progettate un algoritmo efficiente che capisce se hai abbastanza forza per uscire dal labirinto.

¹Scomparire e ricomparire una seconda volta in così breve tempo è faticoso e quindi ti richiede 5 volte più energia che farlo una prima volta.