Cognome: **Nome**: **# Matricola**: **Riga**: **Col**:

Algoritmi e Strutture Dati - 31/10/14

Esercizio 0 Scrivere correttamente nome, cognome, numero di matricola, riga e colonna.

Esercizio 1 – Punti ≥ 6

Si considerino le seguenti equazioni di ricorrenza, per le quali i casi base sono tutti pari a $T(n) = 1$ per $n \leq 1$.

1. $T(n) = T(2n/3) + 2n - 4$
2. $T(n) = 4T(n/2) + n^2\sqrt{n}$
3. $T(n) = 2T(n/4) + \sqrt{n} + 10\log n$
4. $T(n) = 3T(n/2) + 2n\log n + 10n$
5. $T(n) = T(n-6) + n^{5/6}$

Identificare limiti superiori e inferiori per ognuna delle equazioni di ricorrenza (eventualmente stretti, utilizzando la notazione $\Theta(f(n))$), utilizzando un metodo a vostro piacimento. Assumendo che esse provengano dall'analisi di altrettanti algoritmi, quale algoritmo scegliereste?

Esercizio 2 – Punti ≥ 6

Sia dato V un vettore di $n \geq 10^6$ elementi per il quale è noto che i primi $n - \lfloor n^{4/5} \rfloor$ elementi siano già ordinati fra di loro. Scrivere una funzione almostSort(**integer**[ ] V, **integer** n) *in pseudocodice* che ordini l'intero vettore V in tempo inferiore a $\Theta(n \log n)$. Valutare la complessità dell'algoritmo proposto.

Esercizio 3 – Punti ≥ 8

In questo esercizio si consideri un grafo diretto G rappresentato tramite <u>matrice di adiacenza</u>. In questa rappresentazione, è possibile utilizzare le solite primitive G.V() e G.adj(), ma vi ricordo che in questo caso una visita ha costo $O(n^2)$; d'altra parte, l'operazione G.insertEdge(u, v) può essere realizzata in tempo $O(1)$, semplicemente ponendo ad 1 il bit relativo a tale arco.

1. Un grafo diretto G è connesso *debolmente* se per ogni coppia u, v di vertici <u>distinti</u> esiste *almeno una* sequenza di nodi <u>tutti distinti</u> $u = w_1, w_2, \ldots, w_{n-1}, w_n = v$ tale per cui esiste almeno uno degli due archi (w_i, w_{i+1}) o (w_{i+1}, w_i), con $1 \leq i < n$. Scrivere una funzione weaklyConnected(GRAPH G) *in pseudocodice* che determini se il grafo è connesso debolmente oppure no.

2. Un grafo diretto G è connesso *singolarmente* se per ogni coppia u, v di vertici <u>distinti</u> esiste *esattamente una* sequenza di nodi <u>tutti distinti</u> $u = w_1, w_2, \ldots, w_{n-1}, w_n = v$ tale per cui esiste almeno uno degli archi (w_i, w_{i+1}) o (w_{i+1}, w_i). Scrivere una funzione singularlyConnected(GRAPH G) *in pseudocodice* che determini se il grafo è connesso singolarmente oppure no.

Valutare la complessità degli algoritmi proposti.

Esercizio 4 – Punti ≥ 12

In un vettore V di interi, si dice *spessore* del sottovettore $V[i \ldots j]$ la differenza tra il massimo e il minimo valore contenuto nel sottovettore.
Scrivere una funzione thickness(**integer**[ ] V, **integer** n, **integer** C) *in pseudocodice* che, preso un vettore V di n interi ed un intero $C > 0$, restituisca la lunghezza del più lungo sottovettore tra quelli di spessore al più C. Valutare la complessità dell'algoritmo proposto.

Nota: esistono algoritmi con complessità $\Theta(n^3)$, $\Theta(n^2)$, $\Theta(n \log n)$ (basato su divide-et-impera).