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

Algoritmi e Strutture Dati - 28/01/13

Esercizio 1 – Punti ≥ 6 (Parte A)

Supponendo che il caso base sia $O(1)$ si calcoli l'andamento asintotico delle seguenti equazioni di ricorrenza:

1. $A(n) = 4A(n/2) + n^2 \log n$.
2. $B(n) = 4B(n/2) + n^2$.
3. $C(n) = nC(n-1)$.

Esercizio 2 – Punti ≥ 6 (Parte A)

Scrivere un algoritmo efficiente che, dato in input un albero binario T, restituisca **true** se T rappresenta un albero binario completo, **false** altrimenti.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Esercizio 3 – Punti ≥ 9 (Parte A)

Si consideri un grafo non orientato $G = (V, E)$ in cui a ciascun nodo $v \in V$ è associato un peso reale $w(v)$ (che può essere positivo o negativo). Un cammino $\langle v_1, v_2, \ldots, v_k \rangle$ si dice monotono se $w(v1) < w(v2) < \ldots < w(vk)$. In altre parole in un cammino monotono i pesi dei nodi attraversati devono essere in ordine strettamente crescente.

1. Dimostrare che se $\langle v_1, v_2, \ldots, v_k \rangle$ è un cammino monotono, allora è aciclico
2. Scrivere un algoritmo efficiente che, dato in input un grafo non orientato $G = (V, E)$ con nodi pesati, e due nodi $s, d \in V$, restituisca **true** se e solo se esiste un cammino monotono che inizia dalla sorgente s e termina nella destinazione d. L'algoritmo deve anche stampare i nodi che compongono tale cammino (i nodi possono essere stampati nell'ordine $v_1, v_2, \ldots, v_k$ oppure nell'ordine inverso $v_k, v_{k-1}, ...v_1$)
3. Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Esercizio 4 – Punti ≥ 12 (Parte B)

Siano dati un intero positivo T ed un insieme di interi positivi $A = \{a_1, \ldots, a_k\}$. Scrivere un algoritmo basato su programmazione dinamica che ritorni **true** se esiste un sottoinsieme $B \subseteq A$ tale che $T = \sum_{b \in B} b$, **false** nel caso contrario.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.