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

Algoritmi e Strutture Dati – 04/09/17

Esercizio 1 – Punti ≥ 6 (Parte A)

Trovare i limiti superiore e inferiori più stretti possibili per la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 3T(\lfloor n/3 \rfloor) + 6T(\lfloor n/6 \rfloor) + 54T(\lfloor n/12 \rfloor) + n^2 & n > 12 \\ 1 & n \leq 12 \end{cases}$$

Esercizio 2 – Punti ≥ 6 (Parte A)

Sia A un vettore contenente n valori interi (non necessariamente distinti) e si consideri il problema di determinare se esistono o no due valori distinti che compaiono in A lo stesso numero di volte. Per esempio, se $A = [3, 1, 3, 4, 1, 4, 5]$, allora la risposta è **true** (perché 3, 1 e 4 compaiono 2 volte), mentre se $A = [3, 1, 3, 3, 1]$, allora la risposta è **false**. Scrivere un algoritmo che risolva tale problema.

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

Esercizio 3 – Punti ≥ 8 (Parte B)

Andare in snowboard è bellissimo, ma diciamo la verità: i tratti in piano (o peggio ancora, in salita) sono terribili, perchè tocca saltellare come deficienti. Per attirare un maggior numero di snowboarder, una nota località sciistica ha deciso di misurare il percorso più lungo che può essere attraversato senza mai incontrare tratti in piano o in salita.
A questo scopo, ha mappato le proprie piste tramite una serie di punti spaziali. Ha poi identificato le coppie di punti adiacenti, per cui è possibile andare dal punto p_1 al punto p_2 senza dover passare da punti intermedi e senza uscire fuori pista.
Sia quindi $G = (V, E)$ un grafo dove V è l'insieme di punti e E è l'insieme di coppie di punti adiacenti. Sia $d[u][v]$ la distanza fra due punti adiacenti u, v. Sia $z[u]$ l'altitudine del punto u.
Scrivere un algoritmo che restituisce la lunghezza del più lungo cammino $P = u_1, u_2, \ldots, u_k$ tale per cui $\forall i, 2 \leq i \leq k : z[u_{i-1}] > z[u_i]$. La lunghezza totale del cammino è misurata come somma dell distanze dei sui archi:

$$D(P) = \sum_{i=2}^{k} d[u_{i-1}][u_i]$$

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

Esercizio 4 – Punti ≥ 10 (Parte B)

Una stringa w è una *stringa bilanciata di parentesi tonde* se:

- w è una stringa vuota, oppure
- $w = (x)$, dove x è una stringa bilanciata di parentesi tonde, oppure
- $w = xy$, dove x, y sono stringhe bilanciate di parentesi tonde

Ad esempio, "(())" e "()()" sono stringhe bilanciate di parentesi tonde, ")()(" e "()())" non lo sono.
Scrivere un algoritmo che dato in input un intero n, stampi tutte le possibili stringhe bilanciate di parentesi tonde di lunghezza $2n$ (ovvero contenenti n coppie di parentesi).

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

Ad esempio, con $n = 2$, le stringhe da stampare sono "(())" e "()()".