

Cognome Nome:

N. Matricola:

Corso: A B

Esercizio 1. (2+6+3 punti)

Sia A un array **non** ordinato di n interi $A[1], A[2], \dots, A[n]$. Definiamo *salto* in A un indice i , con $1 \leq i < n$, tale che $A[i] - A[i+1] \geq 2$. Si può dimostrare che se $n \geq 2$ e $A[n] - A[1] \geq n$, l'array A ha almeno un salto.

Dato un array A di dimensione n tale che $n \geq 2$ e $A[n] - A[1] \geq n$, si consideri il problema di trovare la posizione i di un salto.

1. Dimostrare che un qualunque algoritmo che risolve il problema suddetto mediante confronti richiede tempo $\Omega(\log n)$ al caso pessimo.
2. Descrivere un algoritmo **ottimo** di tipo *divide-et-impera* per trovare un salto.
3. Calcolare la complessità al caso pessimo dell'algoritmo proposto indicando, e risolvendo, la corrispondente relazione di ricorrenza.

Esercizio 2. (3+3 punti)

Date le stringhe $A = ATGATA$ e $B = AGTACA$, simulare l'algoritmo di programmazione dinamica per il calcolo della Edit Distance tra due stringhe mostrando:

1. Il contenuto della tabella di programmazione dinamica, e il valore della distanza così calcolata;
2. Tutti gli allineamenti ottimi ricostruibili dalla tabella.

Esercizio 3. (8 punti)

Dato un grafo non orientato $G = (V, E)$, un vertice sorgente $s \in V$ e un intero non negativo k , progettare e descrivere in pseudocodice un algoritmo efficiente per stampare i vertici a distanza k dalla sorgente s . Analizzare la complessità dell'algoritmo proposto.

Esercizio 4. (2+2+2 punti)

- 4a. Sia A un array di interi *quasi ordinato*, ovvero A è ordinato fatta eccezione per al più 3 elementi che sono fuori posto. È possibile ordinare A in tempo lineare? Se sì, come? Se no, perché?
- 4b. Si descriva **a parole** (no pseudocodice) il funzionamento di MERGE-SORT e di QUICK-SORT, impiegando al più 5 righe per ciascun algoritmo.
- 4c. Si indichino le complessità di MERGE-SORT e di QUICK-SORT al caso medio, pessimo, e ottimo, indicando per ognuno dei due algoritmi in quali condizioni i tre casi hanno luogo.