

ESAME DI ALGORITMI
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
29 ottobre 2024

Si risolvano i seguenti esercizi in un tempo non superiore a 3 ore. Si abbia cura di consegnare la risoluzione dei primi 4 esercizi in un foglio (FOGLIO A) separato da quello utilizzato per la consegna degli ultimi due esercizi (FOGLIO B). Gli studenti delle vecchie coorti che devono sostenere solo il modulo di Algoritmi dovranno risolvere gli esercizi 1, 2, 3, 5 e 6 (tempo 2 ore). Gli studenti che devono sostenere solo il modulo di Laboratorio dovranno risolvere l'esercizio 4 (tempo un'ora).

——— FOGLIO A ———

1. Fornire le funzioni ricorsive utilizzate per calcolare il costo di una soluzione ottimale negli algoritmi di FLOYD-WARSHALL e ALL-PAIRS-SHORTEST-PATH. In entrambi i casi, spiegare l'idea alla base della funzione ricorsiva e chiarire il significato delle variabili e dei parametri utilizzati nel calcolo.
2. Definire formalmente cosa si intende per *Problema di Ottimizzazione*, evidenziando gli elementi principali che lo distinguono da altri tipi di problemi computazionali. Successivamente, descrivere i concetti fondamentali della *Programmazione Dinamica* come tecnica di risoluzione per problemi di ottimizzazione, spiegandone i principi e le caratteristiche principali.
3. Descrivere formalmente il problema della *parentesizzazione ottimale per la moltiplicazione di una sequenza di matrici*. Spiegare, anche tramite un esempio, perché questo è considerato un problema di ottimizzazione. Fornire la funzione ricorsiva utilizzata per calcolare il costo di una soluzione ottimale e presentare il pseudocodice dell'algoritmo di programmazione dinamica derivato dalla funzione ricorsiva, indicandone la complessità computazionale.
4. Si fornisca lo pseudo-codice (o i codici in linguaggio C/C++) dell'algoritmo COUNTING-SORT per l'ordinamento di un array di valori numerici. Indicare la complessità computazionale dell'algoritmo fornito.

——— FOGLIO B ———

5. Si risolva l'equazione di ricorrenza $T(n) = 2T\left(\frac{n}{3}\right) + \Theta(n)$, utilizzando il metodo preferito e si disegni uno sketch dell'albero di ricorrenza associato all'equazione. Si stabilisca inoltre quale delle seguenti condizioni sono soddisfatte dalla soluzione $T(n)$:
 - $T(n) = \mathcal{O}(n)$;
 - $T(n) = \Omega(n)$;
 - $T(n) = o(n)$;
 - $T(n) = o(n^2)$.
6. Si dimostri formalmente che il tempo computazionale di MAX-HEAPIFY su un albero di dimensione n è $\mathcal{O}(n)$. Si usi inoltre questo risultato per spiegare perché l'algoritmo BUILD MAX-HEAP richiede tempo $\mathcal{O}(n)$.