ESAME DI ALGORITMI
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
24 giugno 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. Si supponga di operare su di un albero Rosso-Nero completo, a meno dell'ultimo livello in cui i nodi sono disposti da sinistra a destra. L'albero contiene 18 chiavi. I nodi dell'albero sono tutti nodi neri ad esclusione dei nodi dell'ultimo livello il cui colore è rosso. Nello specifico si effettuino 6 operazioni di cancellazione della chiave più piccola dell'albero. Si fornisca la configurazione dell'albero dopo ciascuna delle 6 operazioni.

2. Sia $A = [5, 4, 2, 8, 10, 15, 9, 11, 12, 17, 14, 7, 16]$ un array di 13 elementi. Si supponga di eseguire l'**ordinamento inverso** dell'array A mediante l'esecuzione dell'algoritmo HEAPSORT al fine di ottenere un ordinamento non crescente dei suoi elementi. Si fornisca la configurazione dell'array dopo la fase di costruzione dello Heap e dopo ciascuna delle iterazioni dell'algoritmo di ordinamento.

3. Limitandosi a quanto trattato a lezione, elencare gli algoritmi per il calcolo dei **cammini minimi tra tutte le coppie**, indicandone il nome, i limiti applicativi, e la relativa complessitá asintotica. Nel caso in cui l'algoritmo citato sia basato sulla tecnica di programmazione dinamica, si fornisca la funzione ricorsiva utilizzata dall'algoritmo per il calcolo di una soluzione ottima al problema.

4. Si forniscano gli pseudo-codici (o i codici in linguaggio `C/C++`) degli algoritmi per il calcolo dei cammini minimi da sorgente singola BELLMAN-FORD e DIJKSTRA, compresi i codici delle eventuali procedure ausiliarie. Indicare anche la complessità computazionale delle procedure fornite, motivandone la risposta.

——— FOGLIO B ———

5. Si consideri l'equazione di ricorrenza $T(n) = 2T\left(\frac{n}{b}\right) + \sqrt[3]{n}$.
   A. Si risolva l'equazione al variare del parametro reale $b > 1$, utilizzando il metodo Master.
   B. Si stabilisca per quali valori di b la soluzione $T(n)$ all'equazione soddisfa le seguenti condizioni: (i.) $T(n) = \Theta(n)$, (ii.) $T(n) = \Omega(\sqrt[3]{n}\log(n))$, (iii.) $T(n) = o(\sqrt[3]{n}\log(n))$.
   C. Si disegni uno sketch dell'albero di ricorrenza associato all'equazione per $b = 2$.

6. Sia $\star$ l'operazione si numeri natural definita da $a \star b := 2a + 3b$. L'operazione $\star$ non è associative, infatti, ad esempio, $(1 \star 5) \star 2 = 2(2 + 15) + 3 \cdot 2 = 40$ è diverso da $1 \star (5 \star 2) = 2 + 3(10 + 6) = 50$. Ha allora senso considerare il seguente problema di ottimizzazione computazionale.

MAX-$\star$-VALUE PROBLEM
INPUT: $a_1, a_2, \ldots, a_n \in \mathbb{N}$.
GOAL: trovare una parentesizzazione di $a_1 \star a_2 \star \cdots \star a_n$ che ne massimizza il valore (cioè il risultato).

   Si dimostri che MAX-$\star$-VALUE PROBLEM gode della proprietà di sottostruttura ottima.
   Suggerimento: il problema è molto simile a MATRIX-CHAIN MULTIPLY.