ESAME DI ALGORITMI
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
29 gennaio 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, contenente 15 chiavi. I nodi dell'albero sono tutti nodi neri ad esclusione dei nodi del livello 2 il cui colore è rosso. Nello specifico si effettuino 4 operazioni di inserimento di una chiave più grande di quelle presenti nella struttura dati, seguite da 4 operazioni di cancellazione della chiave più piccola contenuta nell'albero. Si fornisca la configurazione dell'albero dopo ciascuna delle 8 operazioni.

2. Si supponga di operare su di un Max-Heap inizialmente vuoto, inserendo le seguenti 13 chiavi, nell'ordine dato: $\langle 2, 5, 4, 7, 9, 8, 10, 15, 11, 12, 17, 14, 16 \rangle$. Si fornisca la configurazione (disegnare l'albero) del Max-Heap dopo ciascuna delle 13 operazioni di inserimento. Indicare quale sarebbe stata la configurazione del Max-Heap nel caso in cui la struttura dati fosse stata costruita mediante l'applicazione della procedura Build-Max-Heap applicata alla medesima sequenza di chiavi.

3. Si consideri una tabella hash ad indirizzamento aperto formata da 10 celle e la cui configurazione, dopo i primi 5 inserimenti è la seguente: `[ X - X X - - X - - X ]`. In tale contesto si supponga che il simbolo `X` rappresenti una cella occupata e che il simbolo `-` rappresenti una cella libera. Si forniscano le probabilità di ciascuna cella di essere scelta come destinataria dalla successiva operazione di inserimento.

4. Si forniscano gli pseudo-codici (o i codici in linguaggio `C/C++`) degli algoritmi Fast-All-Pairs-Shortest-Paths e Floyd-Warshall, utilizzati per il calcolo dei cammini minimi tra tutte le coppie in un grafo orientato e pesato. Fornire inoltre lo pseudo-codice (o il codice) della procedura ricorsiva Print-Shortset-Path(i, j), in grado di stampare il cammino minimo dal nodo i al nodo j. Indicare anche la complessità computazionale delle procedure fornite, motivandone la risposta.

—— Foglio B ——

5. Si consideri l'equazione di ricorrenza $T(n) = aT\left(\frac{n}{4}\right) + 3\sqrt{n}$. Si risolva l'equazione al variare del parametro reale $a \geq 1$, utilizzando il metodo Master e si stabilisca per quali valori di b la soluzione $T(n)$ all'equazione soddisfa le seguenti condizioni

   $$\text{(i.) } T(n) = \Theta(n) \qquad \text{(ii.) } T(n) = \mathcal{O}\left(n^2\right) \qquad \text{(iii.) } T(n) = o\left(\sqrt{n}\log(n)\right).$$

   Infine, si disegni uno sketch dell'albero di ricorrenza associato all'equazione data per $a = 2$.

6. In un grafo non pesato, la lunghezza di un cammino tra due vertici è definita come il numero di archi che compongono il cammino. Si consideri il problema Unweighted Shortest Path, in cui l'input è un grafo $G = (V, E)$ orientato e non pesato, e due vertici distinti $u, v \in V$. L'obiettivo del problema è trovare un cammino di lunghezza minima. Si dimostri che Unweighted Shortest Path ha la proprietà di sottostruttura ottima.