ESAME DI ALGORITMI (SIMULAZIONE)
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
18 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 Min-Heap inizialmente vuoto, inserendo le seguenti 13 chiavi, nell'ordine dato: $\langle 10, 9, 6, 8, 4, 11, 13, 12, 7, 5, 3, 1, 2 \rangle$. Si fornisca la configurazione (disegnare l'albero) del Min-Heap dopo ciascuna delle 13 operazioni di inserimento. Indicare infine quale sarebbe la configurazione della struttura dati dopo un'operazione di estrazione del minimo.

2. 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 1 il cui colore è rosso. Nello specifico si effettuino 6 operazioni di cancellazioni della chiave più piccola contenuta nell'albero e si fornisca la configurazione dell'albero dopo ciascuna delle 6 cancellazioni.

3. Si forniscano le funzioni ricorsive utilizzate per il calcolo del costo di una soluzione ottima utilizzate dagli algoritmi di Programmazione Dinamica per i problemi della Longest Common Subsequence e della Distanza di Editing.

4. Si forniscano gli pseudo-codici (o i codici in linguaggio `C/C++`) degli algoritmi di Bellman-Ford e Dijkstra per la risoluzione dei problemi di cammino minimo da sorgente singola. Indicare anche la complessità computazionale delle procedure fornite, motivandone la risposta.

——— FOGLIO B ———

5. Si consideri l'equazione di ricorrenza $T(n) = 16T(\frac{n}{b}) + \Theta(n^2 \log^2(n))$. Si risolva l'equazione al variare del parametro reale $b > 1$, utilizzando il metodo Master e si stabilisca per quali valori di b la soluzione $T(n)$ all'equazione data soddisfa le seguenti condizioni:

   $$\text{(i.) } T(n) = \mathcal{O}(n^2 \log^2(n)), \qquad \text{(ii.) } T(n) = o(n^2 \log^3(n)), \qquad \text{(iii.) } T(n) = \Theta(n^4).$$

   Si disegni inoltre uno sketch dell'albero di ricorrenza associato all'equazione data per $b = 2$.

6. Si consideri il problema 0-1 KNAPSACK, in cui l'input è un insieme di oggetti $\langle 1, \ldots, n \rangle$ tale che ogni oggetto i abbia un valore v_i e un peso w_i, e un limite superiore W al peso. L'obiettivo di un algoritmo che risolva il problema è trovare un sottoinsieme S di oggetti tale che la somma dei loro valori, $\sum_{i \in S} v_i$, sia massima e la somma dei loro pesi sia limitata da W, cioè $\sum_{i \in S} w_i \leq W$. Si dimostri che 0-1 KNAPSACK ha la proprietà di sottostruttura ottima e si faccia vedere con un controesempio che 0-1 KNAPSACK non ha la propriet$\tilde{\text{A}}$ di scelta greedy.