

ESAME DI ALGORITMI
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
6 settembre 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. Descrivere il processo di cancellazione in un albero rosso-nero, includendo una spiegazione dettagliata dei vari casi gestiti dalla procedura RB-DELETE-FIXUP. Per ciascun caso, fornire esempi grafici che illustrino chiaramente la dinamica dell'operazione.
2. Dato l'array $A = [1, 3, 4, 7, 8, 11, 12, 13, 15, 18, 22, 24, 25]$ contenente 13 elementi, che rappresenta la configurazione iniziale di un MinHeap binario, eseguire 13 operazioni di estrazione del minimo su questa struttura dati. Dopo ciascuna operazione di estrazione, fornire la configurazione dell'array risultante.
3. Si consideri una tabella hash ad indirizzamento aperto formata da $m = 11$ celle e la cui configurazione, dopo i primi 6 inserimenti è la seguente: $[\text{X} \text{ X} \text{ X} \text{ X} - \text{X} - \text{X} - - \text{X}]$. In tale contesto si supponga che il simbolo X rappresenti una cella occupata e che il simbolo $-$ rappresenti una cella libera. Si fornisca la probabilità della cella i -esima, per $0 \leq i < 11$, di essere scelta come destinataria dalla successiva operazione di inserimento. Si supponga che la funzione hash utilizzata per generare la sequenza di scansione di una chiave k sia: $h(k) = (h'(k) + 3i) \bmod (m)$, dove $h'(k)$ è una funzione hash ausiliaria che gode della proprietà di hashing uniforme semplice.
4. Si fornisca lo pseudo-codice (o i codici in linguaggio C/C++) dell'algoritmo HEAPSORT per l'ordinamento in loco di un array di n elementi. Si forniscano anche i codici delle procedure ausiliarie BUILD-MAX-HEAP, EXTRACT-MAX e MAX-HEAPIFY. Indicare anche la complessità computazionale (temporale e spaziale) delle procedure fornite, motivandone la risposta.

——— FOGLIO B ———

5. Si consideri l'equazione di ricorrenza

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(\log(n)).$$

Si risolva l'equazione utilizzando il metodo preferito tra quelli studiati. Si stabilisca, inoltre, quali tra le seguenti condizioni sono soddisfatte dalla soluzione $T(n)$ all'equazione trovata al punto precedente:

- | | |
|--------------------------------------|--------------------------------------|
| (a.) $T(n) = \mathcal{O}(\log^2(n))$ | (b.) $T(n) = \mathcal{O}(\log^3(n))$ |
| (c.) $T(n) = o(\log^2(n))$ | (d.) $T(n) = o(\log^3(n))$ |

Infine, si disegni uno sketch dell'albero di ricorrenza associato all'equazione, indicando il costo del livello i -esimo, l'altezza dell'albero e il numero di foglie.

6. Si consideri una tabella hash a indirizzamento aperto con m slot (o celle o caselle) in cui sono memorizzate n chiavi, con $n < m$. Si indichi qual è il valore atteso del tempo computazionale necessario a una ricerca senza successo nella tabella. Inoltre, si dimostri formalmente la correttezza della risposta alla domanda precedente.