

**ESAME DI ALGORITMI**  
Università degli Studi di Catania  
Corso di Laurea Triennale in Informatica  
**8 luglio 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  $A = [25, 24, 22, 18, 15, 13, 12, 11, 8, 7, 4, 3, 1]$  contenente 13 elementi, che rappresenta la configurazione iniziale di un MaxHeap binario, eseguire 13 operazioni di estrazione del massimo 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: [ X 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 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:  $h(k) = (h'(k) + 2i) \bmod (m)$ , 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 HEAPIFY. Indicare anche la complessità computazionale delle procedure fornite, motivandone la risposta.

—— FOGLIO B ——

5. Si consideri un albero rosso-nero con  $n$  nodi interni dove  $bh(x)$  denota l'altezza nera di un nodo  $x$ .  
(A) Si dimostri che il sottoalbero radicato a un nodo  $x$  contiene almeno  $2^{bh(x)} - 1$  nodi interni.  
(Suggerimento: fare una dimostrazione per induzione sull'altezza di  $x$ .)  
(B) Utilizzando il risultato di A, si dimostri che l'altezza dell'albero rosso-nero è al più  $2 \log(n+1)$ .  
(C) Come possiamo scrivere il valore dell'altezza in notazione asintotica, in base al risultato del punto B?
6. Si fornisca una risposta per i seguenti tre punti:  
(A) Si utilizzi il metodo Master per risolvere la seguente equazione di ricorrenza

$$T(n) = T\left(\frac{2}{3}n\right) + \Theta(1). \quad (1)$$

- (B) L'equazione (1) rappresenta un limite superiore al tempo computazionale di HEAPIFY, ovvero  $T_{\text{Heapify}}(n) \leq T_{\text{Heapify}}\left(\frac{2}{3}n\right) + \Theta(1)$ . Come possiamo allora scrivere il tempo computazionale di HEAPIFY in notazione asintotica?
- (C) Si disegni l'albero di ricorsione di

$$T(n) = 3T\left(\frac{n}{4}\right) + \Theta(n^2), \quad (2)$$

indicando il **costo dell' $i$ -esimo livello, l'altezza e il numero di foglie**. Si usino i risultati per stimare (ovvero fare una guess per) l'ordine di grandezza della soluzione all'equazione (2) in notazione asintotica.