

**ESAME DI ALGORITMI**  
Università degli Studi di Catania  
Corso di Laurea Triennale in Informatica  
**14 febbraio 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 3 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. Sia  $A = [2, 5, 4, 7, 9, 8, 10, 15, 11, 12, 17, 14, 16]$  un array di 13 elementi. Si supponga di eseguire l'ordinamento dell'array  $A$  mediante l'esecuzione dell'algoritmo HEAPSORT al fine di ottenere un ordinamento non decrescente dei suoi elementi. Si fornisca la configurazione dell'array dopo la fase di costruzione dello Heap e dopo ciascuna delle 12 iterazioni dell'algoritmo di ordinamento.
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 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  $T(n) = 3T(\frac{n}{b}) + \Theta(n \log(n))$ .
  - 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.)  $T(n) = \mathcal{O}(n \log^2(n))$       (ii.)  $T(n) = \Theta(n)$       (iii.)  $T(n) = \omega(n \log(n))$ .
  - C. Si disegni uno sketch dell'albero di ricorrenza associato all'equazione per  $b = 3$ , indicando l'altezza dell'albero e il numero di foglie.
6. Si consideri il seguente problema computazionale, denominato ACTIVITY SELECTION PROBLEM.  
**Input:** un insieme  $S = \{a_1, \dots, a_n\}$  di attività, ognuna con uno *starting-time* e un *finish-time*, ovvero  $a_i = [s_i, f_i)$  con  $0 \leq s_i < f_i < +\infty$  per ogni  $1 \leq i \leq n$ .  
**Goal:** selezionare un sottoinsieme di attività  $A \subseteq S$  tale che:
  - le attività in  $A$  siano compatibili, cioè, per ogni  $a_i, a_j \in A$  distinti si abbia  $[s_i, f_i) \cap [s_j, f_j) = \emptyset$ ;
  - $A$  abbia cardinalità massima.Il problema può essere risolto con un approccio greedy, scegliendo ad ogni iterazione l'attività  $a_m$  di  $S_i$  con il più piccolo finish-time (ovvero, l'attività che termina prima) e risolvendo il problema per  $S_{i+1}$ , che è ottenuto da  $S_i$  eliminando  $a_m$  e tutte le attività non compatibili con  $a_m$ . Si dimostri che ACTIVITY SELECTION PROBLEM gode della proprietà della scelta greedy.