

SIMULAZIONE DELL'ESAME DI ALGORITMI

Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
13 giugno 2025

Si risolvano i seguenti esercizi in un tempo non superiore a 3 ore. Si abbia cura di consegnare la risoluzione dei primi 3 esercizi in un foglio (FOGLIO A) separato da quello utilizzato per la consegna degli ultimi 3 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 3 (tempo un'ora).

—— FOGGIO A ——

1. Si consideri un array A di n interi, in cui ogni elemento è compreso tra 0 e k , con $k \geq 0$ noto a priori. Si vuole ordinare A in tempo lineare usando l'algoritmo **CountingSort**.
 - (a) Supponendo $n = 20$ e $k = 5$, riportare lo stato finale dell'array C (array dei conteggi cumulativi) e dell'array ordinato B prodotto da **CountingSort**, dato il seguente input:

$A = [2, 5, 3, 0, 2, 3, 0, 3, 1, 4, 2, 4, 5, 1, 2, 3, 0, 5, 2, 1]$

- (b) Fornire e giustificare la complessità asintotica dell'algoritmo **CountingSort** in funzione di n e k . Discutere in particolare il caso in cui $k = \Theta(n)$ e il caso in cui $k = \Theta(1)$.
2. sia dato un heap binario minimo, in cui ogni nodo è definito dalla seguente struttura:

```
struct Node {  
    int key;  
    struct Node* left;  
    struct Node* right;  
    struct Node* parent;  
};
```

Risolvere i seguenti quesiti considerando uno scenario dove **non si ha accesso** all'array sottostante che normalmente rappresenta l'heap.

- (a) Scrivere una procedura `struct Node* Search(struct Node* root, int k)` che restituisca un puntatore al primo nodo contenente la chiave k , oppure `NULL` se tale chiave non è presente. La procedura deve sfruttare le proprietà dell'heap minimo per evitare visite non necessarie.
 - (b) Analizzare la **complessità nel caso peggiore** della procedura implementata, in funzione del numero di nodi n .
 - (c) È possibile migliorare il **caso medio** rispetto a una visita completa dell'albero? Motivare brevemente.
3. Fornire un esempio concreto di un albero rosso-nero valido contenente esattamente 10 chiavi distinte, in cui:
 - (a) Un'operazione di cancellazione provoca una sequenza di due o più rotazioni, incluso il coinvolgimento del nodo radice.
 - (b) Un'operazione di inserimento viola il vincolo del doppio rosso e viene risolta con una rotazione e un cambio di colore.
 - (c) In entrambi i casi, disegnare l'albero *prima e dopo* l'operazione.

4. Si risolva l'equazione di ricorrenza $T(n) = aT\left(\frac{n}{16}\right) + \sqrt[4]{n} \log^2 n$, al variare del parametro reale $a \geq 1$ utilizzando il metodo Master. Si stabilisca inoltre quale delle seguenti condizioni sono soddisfatte dalla soluzione $T(n)$:
- (i) $T(n) = o(n^2)$;
 - (ii) $T(n) = \Theta(\sqrt{n})$;
 - (iii) $T(n) = \Omega(n \log n)$.
5. Si mostri che il problema MATRIX CHAIN MULTIPLY gode della proprietà di sottostruttura ottima.
6. Si scriva la formula ricorsiva utilizzata dall'algoritmo FLOYD-WARSHALL e si simuli tale algoritmo per trovare la tabella (matrice) dei cammini minimi tra tutte le coppie di vertici del grafo in figura di cui si richiede la matrice di adiacenza.

