

**Esame scritto ASD settembre 2023**  
**Tempo totale 50 minuti; punteggio massimo 6.9; sufficienza 4 punti**

**Risposte prive di motivazione chiara e convincente comportano l'assegnazione di 0 punti**  
**Le risposte sulla complessità degli algoritmi saranno valutate solo se la descrizione del relativo algoritmo è corretta**

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

Matricola \_\_\_\_\_

**Hai fatto richiesta ai docenti di agevolazioni per DSA?** Sì ☐ No ☐

Se hai fatto correttamente richiesta, rispondi solo alle **domande 1 e 2**

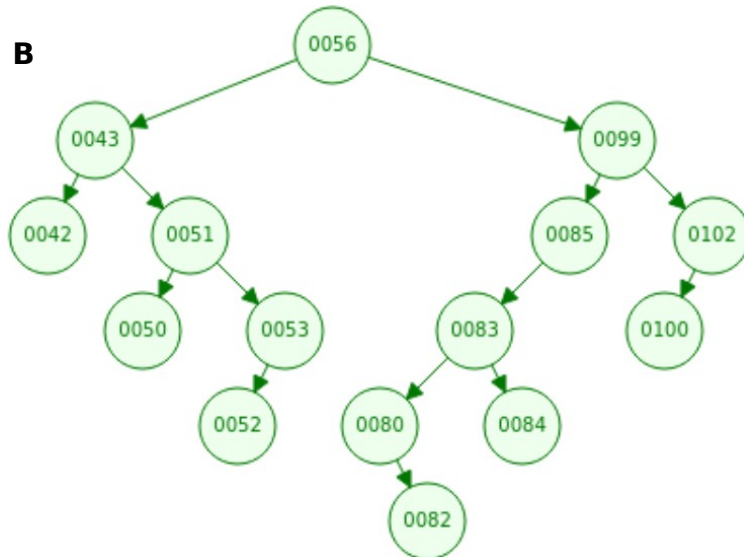
**Domanda #1, quicksort**

**[1/2 punteggio, quicksort caso migliore]** Si scriva quanto vale la complessità di quicksort nel caso **migliore** e si illustri dettagliatamente **quando** si ricade nel caso **migliore** e **come** si calcola la complessità di quicksort nel caso **migliore**.

**[1/2 punteggio, quicksort caso peggiore]** Si simuli l'esecuzione di quicksort sulla sequenza riportata sotto scegliendo il pivot in modo che si ricada nel caso **peggiore**.

82    17    33    29    23    41    59    16

**Domanda #2, binary search tree**



**[Domanda 2.1]** Si descriva in modo generale - cioè senza fare riferimento all'esempio riportato a fianco - l'algoritmo deleteElem su Binary Search Tree. Se l'algoritmo fa uso di funzioni ausiliarie, si spieghi anche il loro funzionamento.

**[Domanda 2.2 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D2.1 è corretto)]** Si indichi la complessità di deleteElem nel caso peggiore motivando la risposta.

**[Domanda 2.3]** Si spieghino dettagliatamente, mediante testo e disegni chiari, i passaggi della chiamata deleteElem(B, 0056); incluse le eventuali chiamate a funzioni ausiliarie, effettuata sul BST **B** disegnato a fianco.

### Domanda #3, tabelle di hash

Si consideri una tabella di hash con liste di collisione che implementa un dizionario **D** in cui le chiavi sono sequenze di 4 cifre decimali che indichiamo con **a, b, c, d** e i valori sono stringhe.

La tabella di hash ha 7 bucket indicizzati da 0 a 6 e la funzione di hash è **h:  $(a^2 + b + c + d) \bmod 7$** .

**[Preparazione dei dati, senza voto MA LA PRESENZA DI DUE O PIU' ERRORI NEL CALCOLO DELLA FUNZIONE DI HASH COMPORTA L'ASSEGNAZIONE DI ZERO PUNTI ALL'INTERO ESERCIZIO]**

**Completate su questo foglio** lo schema mostrato sotto calcolando la funzione di hash delle chiavi scritte nella prima riga e riportandola nella corrispondente cella nella terza riga:

chiave = 3 7 4 1	chiave = 1 1 4 7	chiave = 2 2 7 1	chiave = 7 2 5 2	chiave = 2 1 6 3	chiave = 7 2 0 8	chiave = 1 3 1 1	chiave = 2 7 8 9
valore = "ma"	valore = "co"	valore = "gra"	valore = "pe"	valore = "le"	valore = "re"	valore = "ca"	valore = "giu"
h(chiave) =	h(chiave) =	h(chiave) =	h(chiave) =	h(chiave) =	h(chiave) =	h(chiave) =	h(chiave) =

**[Domanda 3.1, 1/3 del punteggio]** Il dizionario **D** implementato dalla tabella di hash è inizialmente vuoto.

**D3.1:** Disegnate la tabella di hash che si ottiene dagli inserimenti delle stringhe “ma”, “co”, “gra”, “pe”, “le”, “re”, “ca”, “giu” associate alle chiavi su quattro cifre riportate nello schema sopra, nell'ordine in cui compaiono nello schema da sinistra verso destra. Nel caso ci fossero delle chiavi duplicate, ignorate l'inserimento e scrivete esplicitamente sul foglio “questa coppia chiave-valore è stata ignorata perché la chiave è duplicata”.

**[Domanda 3.2, 1/3 del punteggio (la risposta sarà valutata solo se la risposta alla domanda 3.1 è corretta)]**

**D3.2.1:** Quali sono le due principali “buone proprietà” che una funzione di hash deve avere?

**D3.2.2:** La funzione di hash **h** applicata alle chiavi nello schema disegnato sopra gode della prima di queste due “buone proprietà”? Motivare esaurientemente la risposta.

**D3.2.3:** La funzione di hash **h** applicata alle chiavi nello schema disegnato sopra gode della seconda di queste due “buone proprietà”? Motivare esaurientemente la risposta.

**[Domanda 3.3, 1/3 del punteggio]** Sia **m** la dimensione della tabella e sia **n** il numero di elementi nella tabella. Si assuma che **m** ed **n** siano entrambi diversi da 0.

**D3.3.1:** Si descriva l'algoritmo per l'inserimento di un elemento data la sua chiave; l'algoritmo deve esplicitamente e correttamente gestire il caso di chiavi duplicate.

**D3.3.2 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto):** Sotto quali condizioni l'inserimento ha complessità  $\Theta(n)$ ?

**D3.3.3 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto):** Sotto quali condizioni l'inserimento ha complessità  $\Theta(n/m)$ ?

**D3.3.4 (la risposta sarà valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto):** Sotto quali condizioni l'inserimento ha complessità  $\Theta(1)$ ?