

esame 12 luglio 2024

martedì 13 agosto 2024 11:27

[ABBACADA] Esame scritto ASD 12 luglio 2024
Tempo totale 60 minuti; punteggio massimo 6,9; sufficienza 4 punti
Risposte prive di motivazione chiara e convincente comportano l'assegnazione di 0 punti

Nome _____
Cognome _____
Matricola _____

Hai fatto richiesta di agevolazioni per DSA/BES, ricevendo conferma da Di Domizio e Mascardi?
Se hai fatto correttamente richiesta, rispondi solo a due domande a tua scelta; avrai 60 minuti per svolgerle. Se sei anche iscritto al Patto e non ti sei presentato a giugno avrai 90 minuti.

Sei iscritto al Patto e a giugno non ti eri presentato allo scritto?
Se sì, rispondi solo a due domande a tua scelta; avrai 60 minuti per svolgerle (ma solo se a giugno non ti eri presentato).

Se hai risposto no a entrambe le domande sopra, svolgi l'esame completo in 60 minuti.

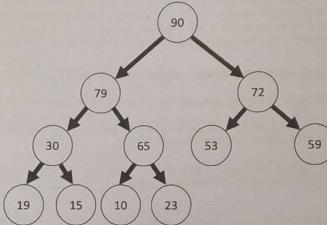
Domanda #1, heap binari

[Domanda 1.1] Si descriva in modo chiaro e generale - cioè, senza nessun riferimento all'esempio di fianco - l'algoritmo deleteMax su heap binari di tipo max.

[Domanda 1.2] Si spieghi la complessità di deleteMax nel caso peggiore, fornendo una motivazione chiara e convincente.

[Domanda 1.3] Si spieghino dettagliatamente, mediante testo tecnicamente corretto in ogni affermazione e disegni chiari (dovete fornire sia testo che disegni), tutti i passaggi della chiamata `deleteMax(A);` effettuata sullo heap A disegnato a fianco.

46 22 93 81 34 16 57 66



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 indicano con a, b, c, d i valori sono stringhe.

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

[Preparazione dei dati, senza voto MA LA PRESENZA DI DUE O PIU' ERORI 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 5 2	chiave = 3 1 4 7	chiave = 2 2 7 1	chiave = 7 2 9 7	chiave = 9 6 6 2	chiave = 4 2 0 7	chiave = 3 3 1 3	chiave = 1 7 8 7
valore = "ma"	valore = "co"	valore = "gra"	valore = "pe"	valore = "le"	valore = "re"	valore = "ca"	valore = "giu"
$h(\text{chiave}) =$							

[Domanda 3.1] 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 (prima si inserisce "ma"; poi "co"; ecc.). Nel caso ci fossero delle chiavi duplicate, ignorate l'inserimento e scrivete esplicitamente sul foglio "coppia chiave-valore ignorare perché la chiave è duplicata".

[Domanda 3.2 (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"? Motivate 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"? Motivate esaurientemente la risposta.

[Domanda 3.3] 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 singolo elemento data la sua chiave; l'algoritmo deve esplicitamente e correttamente gestire il caso di chiavi duplicate.

D3.3.2 (valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto): Sotto quali condizioni l'inserimento ha complessità Theta(1)?

D3.3.3 (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 (valutata solo se l'algoritmo descritto nella risposta D3.3.1 è corretto): Sotto quali condizioni l'inserimento ha complessità Theta(1)?

Domanda 1.1

DeleteMax eseguito su heap binari di tipo max si esegue cancellando il nodo della radice dell'heap, in quanto sempre il numero maggiore.

Domanda 1.2

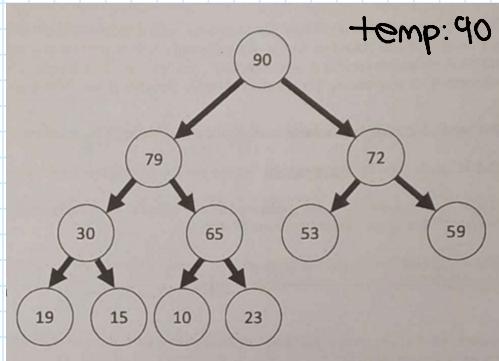
La complessità è $O(\log N)$, in quanto complessità swap padre-figlio + complessità swap radice-ultimo nodo

$$O(1) + O(H) = O(H)$$

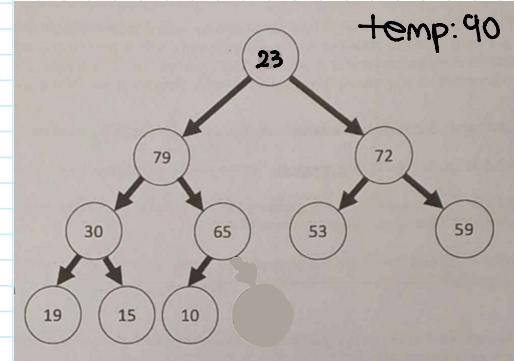
e l'altezza $H = O(\log N)$, con N che rappresenta il numero di nodi.

Domanda 1.3

1) Salvare l'elemento nel nodo della radice

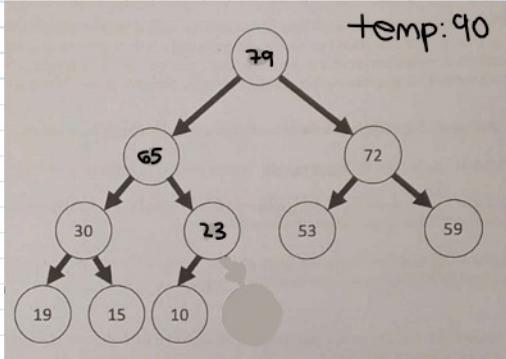
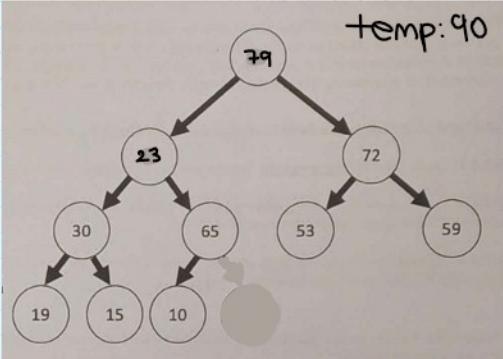


2) Sposto l'ultimo nodo disponibile nella radice

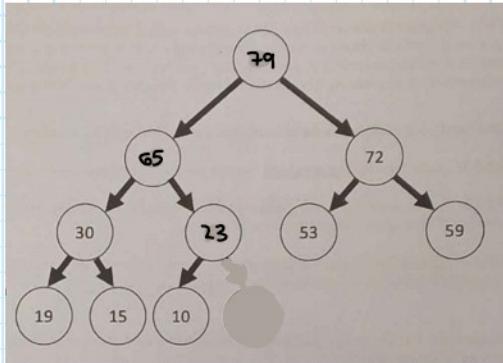


3) visto che il figlio di 23 è maggiore di quest'ultimo, swapo

4) entrambi i numeri sono maggiori, quindi swapo con il più grande



5) abbiamo finito perché il figlio è più piccolo, quindi ritorniamo temp



Domanda 2.1

Complessità: $O(n \log n)$

Il caso migliore è un caso ipotetico dove usiamo come pivot il mediano.

L'array è diviso in due sottosequenze di lunghezza circa $\frac{n}{2}$, ottenendo quindi un albero binario di altezza $\log_2(n)$. Ad ogni livello J la partition viene effettuata su $(2^J) \cdot (\frac{n}{2}^J)$ elementi e ha quindi complessità $O(n)$. Dato che l'altezza ricorsivo è $\log n$, e ogni livello ha un costo di $O(n)$, il costo complessivo nel caso migliore è

$$O(n) + O\left(\frac{n}{2}\right) + O\left(\frac{n}{4}\right) + \dots + O(1) = O(n \cdot \log n)$$

Domanda 2.2

blu: pivot

rosso: elementi in posizione finale

46 22 93 81 34 16 57 66
16 46 22 93 81 34 57 66
16 22 46 93 81 34 57 66
16 22 34 46 93 81 57 66
16 22 34 46 57 93 81 66
16 22 34 46 57 66 93 81
16 22 34 46 57 66 81 93

Domanda 3.1

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

$$(a + b + c \cdot d) \bmod 7.$$

$$3752 = 3 + 5 \cdot 2 = 3 + 10 = 13$$

$$13/7 = 1 \text{ con resto } 6$$

$$\begin{array}{r} 3147 = 3 + 1 + 4 = 8 \\ 8 \mid 7 = 1 \text{ con resto } 1 \end{array}$$

$$\begin{array}{r} 2271 = 2+2+1=5 \\ 517 = 0 \text{ con resto } 5 \end{array}$$

$$7297 = 2 + 9 = 11$$

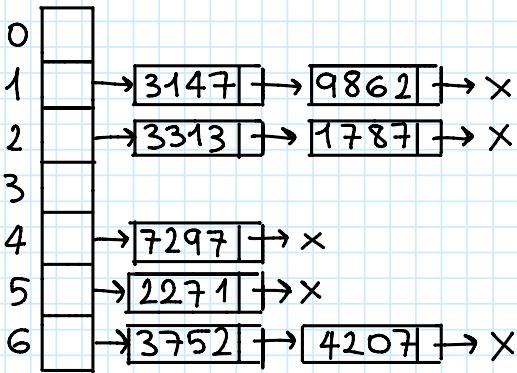
$$11 / 7 = 1 \text{ resto } 4$$

$$\begin{array}{r} 9862 \\ \times 29 \\ \hline 2917 \text{ rest } 1 \end{array}$$

$$4207 = 4+2 = 6$$

$$6/7 = 0 \text{ resto } 6$$

$$\begin{array}{r} 3313 = 3+3+1 \\ 9/7=1 \text{ resto } 2 \end{array}$$



Domanda 3.2.1

Le due principali buone proprie' che una funzione di hash deve avere SONO:

- essere calcolabile in tempo costante
 - essere uniforme, ossia avere tutte le chiavi distribuite in maniera equa nel bucket

Domanda 3.2.2

La prima proprietà è soddisfatta perché la funzione di hash richiede operazioni aritmetiche semplici (somma, moltiplicazione e modulo), ossia a tempo costante in quanto il tempo di esecuzione non varia dall'input. Inoltre, tutte le chiavi sono sequenze di 4 cifre che vanno da 0 a 9.

Domanda 3.2.3

La seconda proprietà non viene soddisfatta in quanto alcuni bucket contengono più chiavi di altri: questo perché alcune chiavi generano più frequentemente lo stesso valore di hash.

Domanda 3.3.1

L'algoritmo di inserimento calcola $h(\text{chiave})$, cerca se in quel bucket è presente e se non lo è la inserisce.

Domanda 3.3.2

L'inserimento ha complessità theta(n) quando la funzione di hash distribuisce tutte le chiavi in un solo bucket

Domanda 3.3.3

L'inserimento ha complessità theta(n/m) quando la funzione di hash distribuisce le chiavi in maniera uniforme

Domanda 3.3.4

L'inserimento ha complessità theta(1) quando la funzione di hash distribuisce le chiavi in maniera iniettiva, o quando M è maggiore di n di molto, portando ad una bassa probabilità di collisioni.