

7 luglio 2021

mercoledì 21 agosto 2024 17:50

Esame ASD 7 luglio 2021 - 3 domande (gli studenti/studentesse con certificazione DSA o che sostengono l'esame in modalità integrata devono fare solo le prime due); tempo totale 45 minuti; punteggio massimo 6.9; sufficienza 4 punti

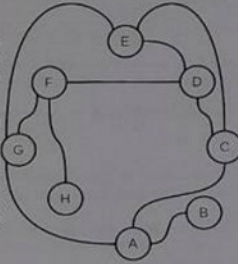
Domanda #1 (per tutti)

Si consideri il grafo non orientato e non pesato disegnato a lato:

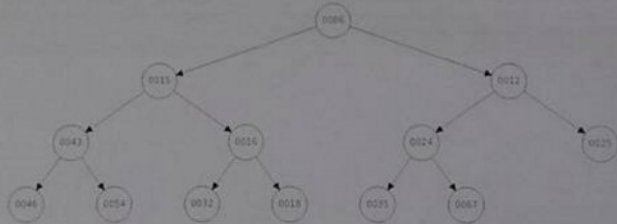
[2/3 del punteggio] Si illustri mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita DFS di tale grafo che parta dal nodo etichettato con H con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

Risposte generiche e non riferite a questo grafo saranno considerate insufficienti.

[1/3 del punteggio] Se è possibile, si disegni un albero di ricoprimento differente da quello ottenuto al passo precedente, sempre generato da una visita DFS del grafo che parta dal nodo H (senza illustrare i passaggi con cui viene generato). Se non è possibile ottenere un albero di ricoprimento diverso, se ne spieghi la ragione.



Domanda #2 (per tutti)



In uno Heap Binario di tipo "min" la radice ha chiave minima e ogni nodo discendente di un nodo N deve avere etichetta maggiore di quella di N: le operazioni sono praticamente identiche a quelle di uno Heap Binario di tipo max come visto a lezione, invertendo "<" e ">".

Si consideri lo Heap Binario di tipo min disegnato sopra, che indicheremo con A.

[1/3 del punteggio] Assumendo che le chiavi siano numeri interi, si disegni come viene modificato lo heap A dopo la seguente chiamata (senza fornire alcuna spiegazione dei passaggi: disegnate solo il risultato)

`insert(4, "elem", A);`

La chiamata `insert(4, "elem", A)` rappresenta il caso peggiore della operazione insert sullo heap A, rispetto alla complessità temporale? Motivare la risposta.

[2/3 del punteggio] Si spieghino dettagliatamente, mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata

`deleteMin(A);`

effettuata sullo heap A modificato a seguito dell'inserimento dell'elemento con chiave 4.

La chiamata `deleteMin(A)` rappresenta il caso peggiore della operazione deleteMin sullo heap A, rispetto alla complessità temporale? Motivare la risposta.

Risposte generiche e che non fanno riferimento allo heap A verranno considerate insufficienti.

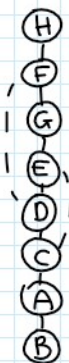
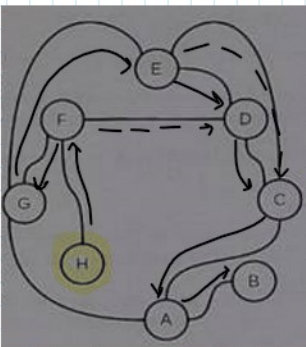
Domanda #3 (se siete in possesso di una certificazione DSA oppure se state svolgendo lo scritto in modalità integrata non dovete rispondere a questa domanda)

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

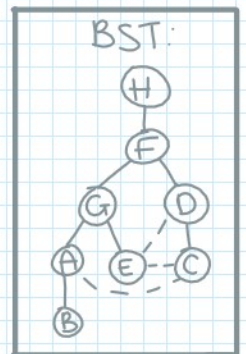
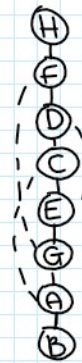
[1/2 punteggio] Si simuli una esecuzione della chiamata di quicksort sulla sequenza riportata sotto, con il pivot scelto sempre, "per magia", come elemento mediano della sottosequenza da riordinare. Questa situazione coincide con il caso migliore di quicksort, con il caso peggiore, o con il caso medio? Qual è la complessità di quicksort in questa specifica situazione?

70 10 20 30 40 90 60 80 100

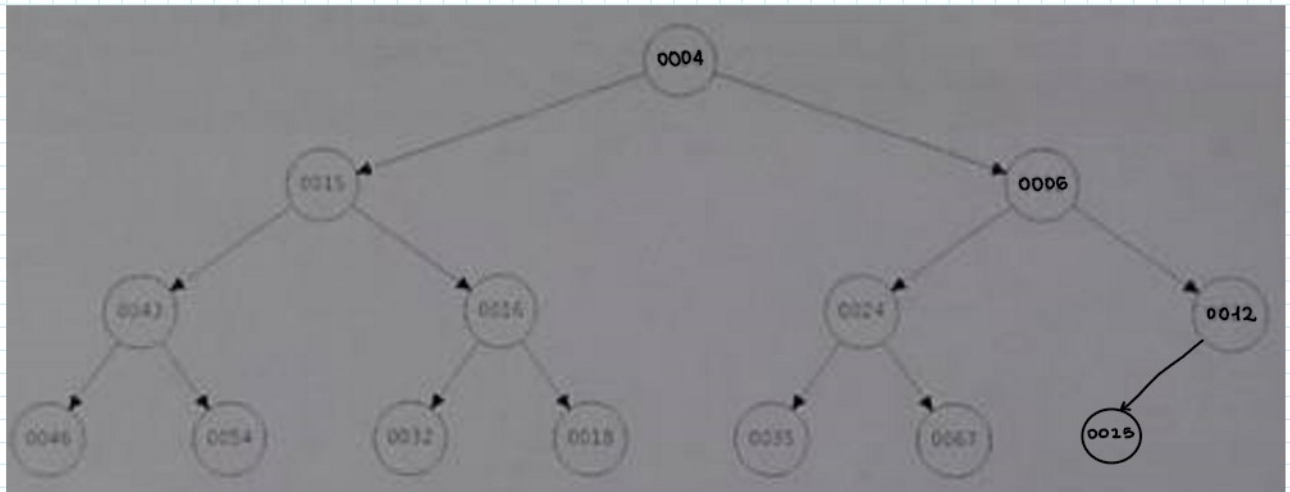
Domanda 1.1



Domanda 1.2



Domanda 2.1



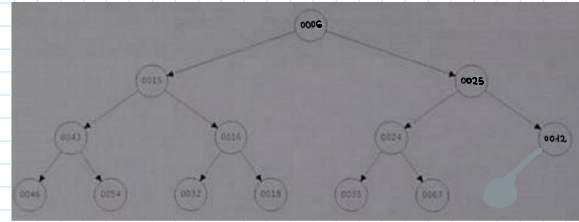
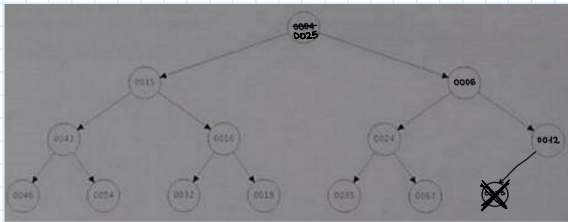
Ci troviamo nel caso peggiore in quanto dobbiamo risalire l'heap fino alla radice. La complessità dell'aggiungere un elemento è $O(1)$, mentre lo swap è $O(1)$. Visto che arriviamo alla radice e l'altezza dell'albero è $\log n$, la complessità dell'inserimento è $\Theta(\log n)$.

dell'aggiungere un elemento è $O(1)$, mentre lo swap è $O(1)$. Visto che arriviamo alla radice e l'altezza dell'albero è $\log n$, la complessità dell'inserimento è $\Theta(\log n)$

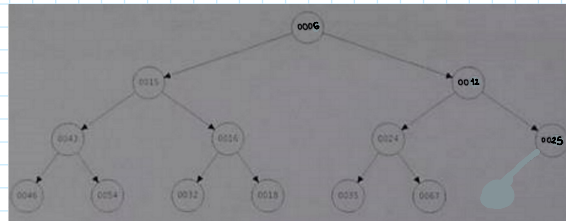
Domanda 2.2

1) Swappo 4 con 25 e elimino 4

2) confronto 25 con 15 e 6, e swappo col minore



3) confronto e swappo 25 con 12



Domanda 3.1

Nel caso peggiore, quicksort ha complessità $O(n^2)$ e si ricade in questo caso quando si sceglie come pivot l'elemento minore o maggiore della sequenza. Per calcolare la complessità si sommano le operazioni fatte ad ogni chiamata ricorsiva. La "prima volta" si eseguono n operazioni, la seconda volta $n-1$ operazioni e così via fino all'ultimo livello in cui si esegue 1 operazione. Questo calcolo si sviluppa nella sommatoria per "i" che va da 1 a n . Quindi $n + (n-1) + (n-2) + \dots + 1 = n^2$.

Domanda 3.2

70 10 20 30 40 90 60 80 100
 10 20 30 40 60 70 90 80 100
 10 20 30 40 60 70 80 90 100
 10 20 30 40 60 70 80 90 100

Ci troviamo nel caso migliore e la complessità è $\Theta(n \log n)$