

Esame ASD 27 luglio 2021 - 3 domande (gli studenti/studentesse con certificazione DSA o che sostengono l'esame in modalità integrata devono fare solo le prime due e **DEVONO SCRIVERE SUL FOGLIO, ACCANTO AL NOME, "DSA" oppure "INTEGRATA"**); tempo totale 45 minuti; punteggio massimo 6.9; sufficienza 4 punti

Domanda #1 (per tutti)

Si consideri una tabella di hash con liste di collisione che implementa un dizionario D in cui le chiavi sono stringhe di 4 cifre decimali che indichiamo con c1, c2, c3, c4 e i valori sono stringhe. La tabella di hash ha 7 bucket indicizzati da 0 a 6 e la funzione di hash è $h: (c1+c2+c3+c4) \bmod 7$.

Passo 1 (senza voto): scrivete sul foglio della risposta le cifre 0 4 0 1 4 0, poi la vostra matricola universitaria (7 cifre), poi la vostra matricola universitaria al contrario (7 cifre), poi la vostra data di nascita nel formato ANNO (4 cifre) MESE (2 cifre) GIORNO (2 cifre), e infine le cifre 1 3 0 3

Ad esempio, se la vostra matricola è 1234567 e siete nati il primo gennaio 2000, dovreste scrivere sul foglio

0 4 0 1 4 0	1 2 3 4 5 6 7	7 6 5 4 3 2 1	2 0 0 0 1 0 1	1 3 0 3
0 4 0 1 4 0	matricola	matricola al contrario	data nascita	1 3 0 3

Passo 2 (senza voto): disegnate sul foglio della risposta uno schema come quello mostrato sotto e immettete i dati indicati:

- nella prima riga, il numero che avete generato al Passo 1, diviso in gruppi di quattro cifre: questi gruppi rappresenteranno le chiavi
- nella seconda riga, i valori associati alle chiavi, ovvero le stringhe "amo", "barca", "cesto", "daino", "edera", "foglia", "gatto", "hotel"
- nella terza riga, il valore di hash associato a ogni chiave, che **dovrete calcolare** usando la formula $h: (c1+c2+c3+c4) \bmod 7$

0401	4012	3456	7765	4321	2000	0101	1303
amo	barca	cesto	daino	edera	foglia	gatto	hotel
val. di h(0401)	val. di h(4012)	val. di h(3456)

Passo 3 (1/3 del punteggio): Il dizionario D implementato dalla tabella di hash è inizialmente vuoto. **Disegnate** la tabella di hash che si ottiene dagli inserimenti delle stringhe "amo", "barca", "cesto", "daino", "edera", "foglia", "gatto", "hotel" associate alle chiavi su quattro cifre così come le avete calcolate al Passo 2, nell'ordine in cui le avete scritte da sinistra verso destra. Ricordate che in un dizionario non si possono ammettere chiavi duplicate: nel caso ce ne fossero ignorate l'inserimento del duplicato e scrivete esplicitamente sul foglio "questa coppia chiave-valore è stata ignorata perché la chiave è duplicata". Continuando l'esempio, si dovrebbe disegnare la tabella di hash risultante da:

```
insert(D, "0401", "amo").
insert(D, "3012", "barca").
insert(D, "3456", "cesto").
.....
```

Passo 4 (1/3 del punteggio): quali sono le due principali "buone proprietà" che una tabella di hash deve avere? Rispetto alle chiavi che avete ottenuto al Passo 2.1, la funzione di hash h ha queste due "buone proprietà"? Motivate esaurientemente la risposta. **Risposte generiche e non relative alla funzione h applicata alle chiavi così come ottenute al Passo 2.1 comportano l'assegnazione di 0 punti.**

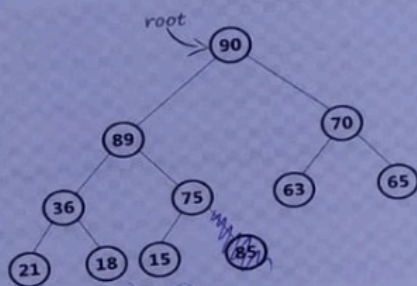
Passo 5 (1/3 del punteggio): sotto l'ipotesi che la funzione di hash abbia tutte le "buone proprietà" che dovrebbe avere, qual è la complessità dell'operazione di **cancellazione** di un elemento data la sua chiave nel caso peggiore? Non basta

Stamp

copiare la risposta dalle slide del corso o da altro materiale didattico: è necessario spiegare cosa rappresentano i parametri che caratterizzano la risposta e motivarla esaurientemente.

La mancanza di motivazione chiara ed esauriente comporta l'assegnazione di 0 punti a questa domanda.

Domanda #2 (per tutti)



Si consideri lo Heap Binario di tipo max (la radice immagazzina il valore massimo) disegnato sopra, che indicheremo con A.

[1/2 del punteggio] Si spieghino dettagliatamente, mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata `insert(86, "elem", A);` effettuata sullo heap A. La chiamata `insert(86, "elem", A)` rappresenta il caso peggiore della operazione insert sullo heap A, rispetto alla complessità temporale? Motivare la risposta.

[1/2 del punteggio] Si spieghino dettagliatamente, mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata `deleteMax(A);` effettuata sullo heap A modificato a seguito dell'inserimento dell'elemento con chiave 86.

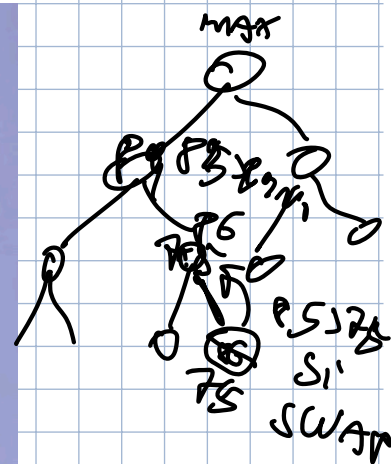
Risposte generiche e che non fanno riferimento allo heap A verranno considerate insufficienti.

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?

1 100 2 90 3 80 70 60 50



NO caso
peggiore
che solo 2
scambi

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

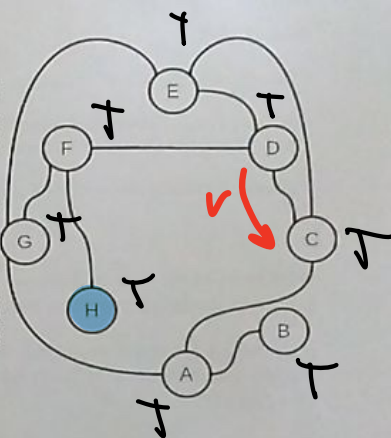
Domanda #1 (per tutti) ✓

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

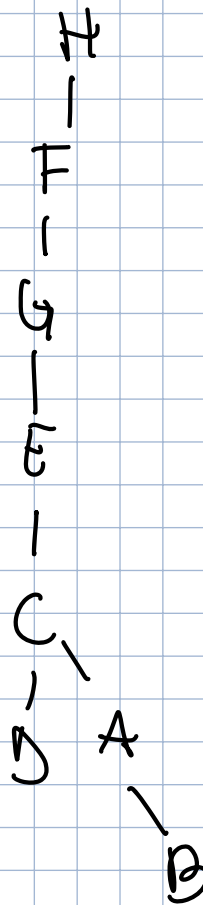
[2/3 del punteggio] Si illustrino mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita DFS di tale grafo che parta dal nodo etichettato con **H** con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

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

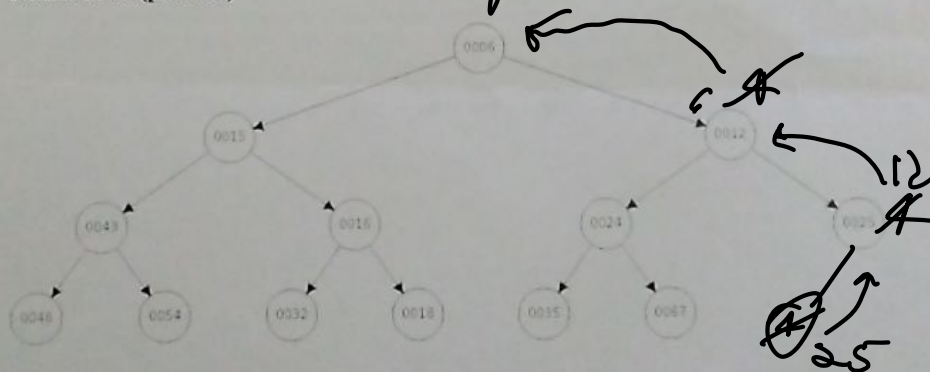
[1/3 del punteggio] Se è possibile, si disegni un albero di ricoprimento differente da quello ottenuto al passo precedente, sempre generato da una visita DFS del grafo che parta dal nodo **H** (senza illustrare i passaggi con cui viene generato). Se non è possibile ottenere un albero di ricoprimento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



ALBERO 1



Domanda #2 (per tutti)



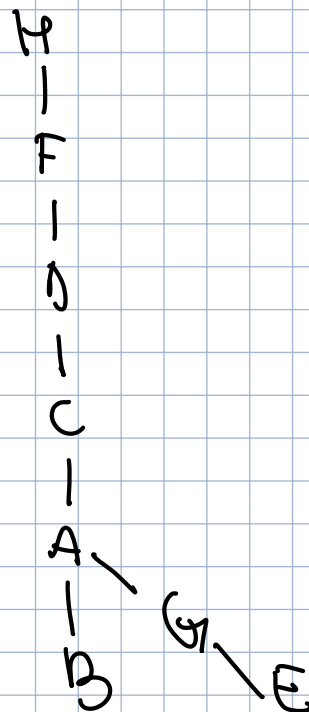
In uno Heap Binario di tipo "min" la radice ha chiave minima e ogni nodo discendente di un nodo N deve avere etichetta maggiore di quella di N: le operazioni sono praticamente identiche a quelle di uno Heap Binario di tipo max come visto a lezione, invertendo "<" e ">".

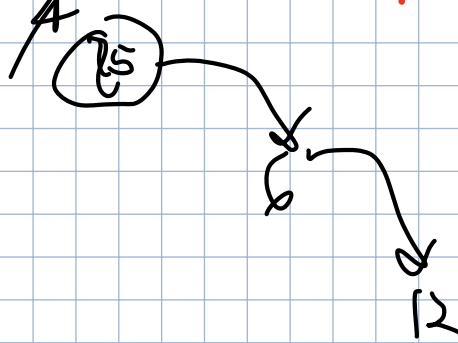
Si consideri lo Heap Binario di tipo min disegnato sopra, che indicheremo con A.

[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);

ALBERO 2





$h' = \text{lunghezza albero}$

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

$h' = \log(\text{numero di nodi})$

[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

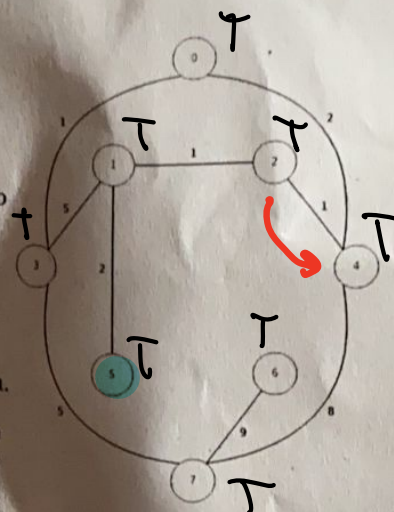
Domanda #1 ✓

Si consideri il seguente grafo pesato e non orientato:

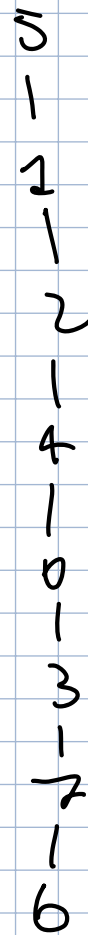
[2/3 del punteggio] Si illustrino mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita DFS (non importa se implementata mediante algoritmo ricorsivo o iterativo: la visita deve sempre andare in profondità) di tale grafo che parta dal nodo etichettato con 5 con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

Non è richiesto di illustrare la rappresentazione del grafo usando qualche struttura dati specifica: la spiegazione dei passi dell'algoritmo deve prescindere da come il grafo verrà rappresentato in un programma.

[1/3 del punteggio] Se è possibile, si disegni un albero di ricoprimento differente da quello ottenuto prima, sempre generato da una visita DFS del grafo che parta dal nodo 5 (senza illustrare i passaggi con cui viene generato). Se non è possibile ottenere un albero di ricoprimento diverso, se ne spieghi la ragione.



ALBERO 1



Domanda #2

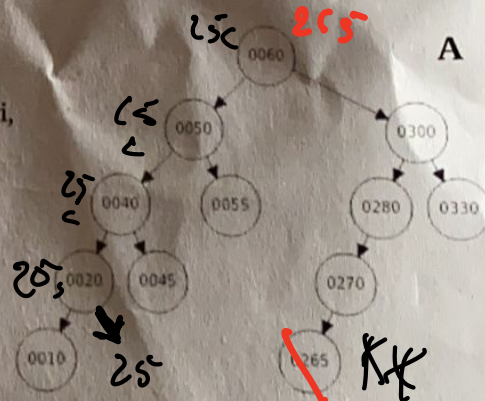
Si consideri il seguente BST che indicheremo con A.

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

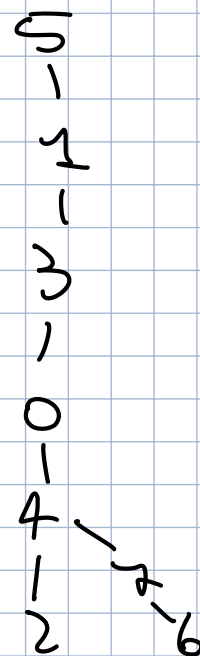
`insertElem(25, "elem", A);`

La chiamata `insertElem(25, "elem", A)` rappresenta il caso peggiore della operazione `insertElem` sui BST, rispetto alla complessità temporale? Motivare la risposta.

SI Ogn



ALBERO 2



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

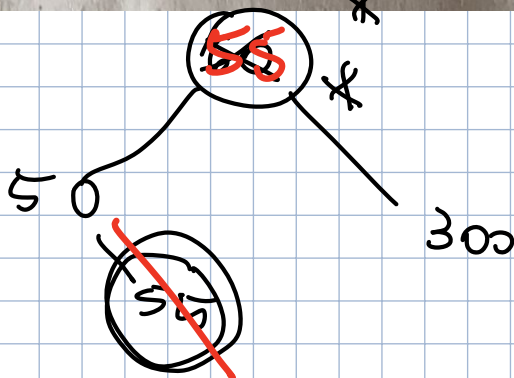
`deleteElem(60, A);`

effettuata sull'albero A modificato a seguito dell'inserimento dell'elemento con chiave 25.

La chiamata `deleteElem(60, A)` rappresenta il caso peggiore della operazione `deleteElem` sui BST, rispetto a complessità temporale? Motivare la risposta.

NO ≠ Oh

* & Si Oh



Domanda #3 (se siete in possesso di una certificazione DSA non dovete rispondere a questa domanda)

[1/2 punteggio] 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] Si simuli una esecuzione della chiamata di quicksort sulla sequenza riportata sotto, con il pivot scelto sempre come ultimo elemento 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?

10 20 30 40 50 60 70 80 90 100

La complessità di Quicksort nel caso migliore vale $O(n \log n)$.

Il caso migliore è un caso ipotetico poiché per ricaderci bisogna ipotizzare che il pivot scelto sia sempre l'elem. mediano

Domanda #1 (per tutti)

[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

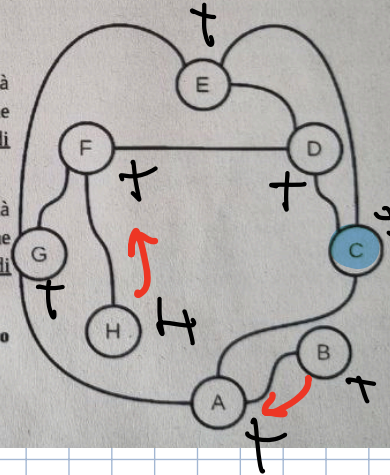
Domanda #2 (per tutti)

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

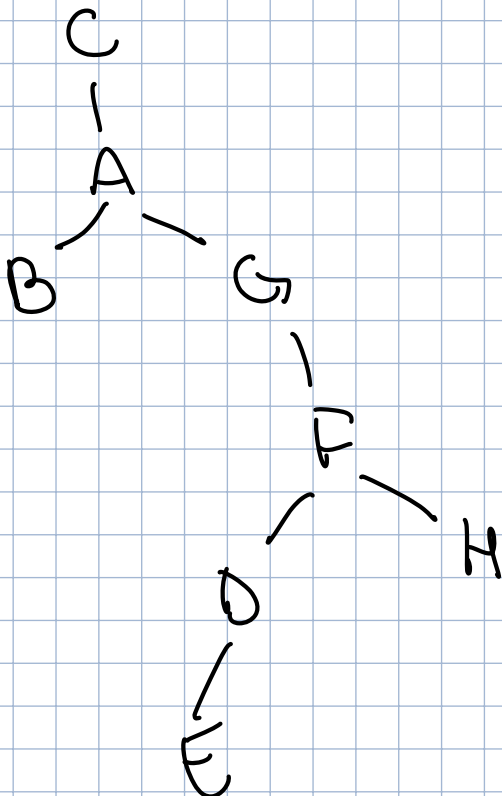
[1/2 del punteggio] Si illustrino mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita DFS di tale grafo che parta dal nodo etichettato con C con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

[1/2 del punteggio] Si illustrino mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita BFS di tale grafo che parta dal nodo etichettato con B con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

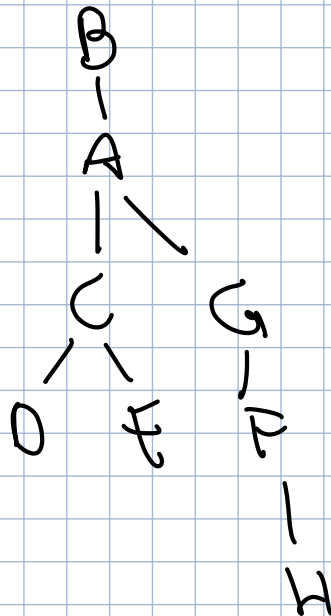
Risposte generiche e non riferite a questo grafo saranno considerate insufficienti.



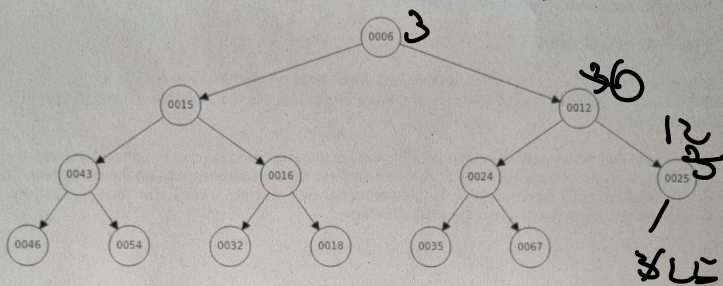
DFS



BFS



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)



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(3, "elem", A);`

La chiamata `insert(3, "elem", A)` rappresenta il caso peggiore della operazione `insert` sullo heap A, rispetto alla complessità temporale? Motivare la risposta.

si → $O(\log n)$

[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 3 (si noti che se lo svolgimento della prima domanda è errato, non sarà possibile valutare questa seconda domanda: controllare la correttezza dell'inserimento).

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

si → $O(\log n)$

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

