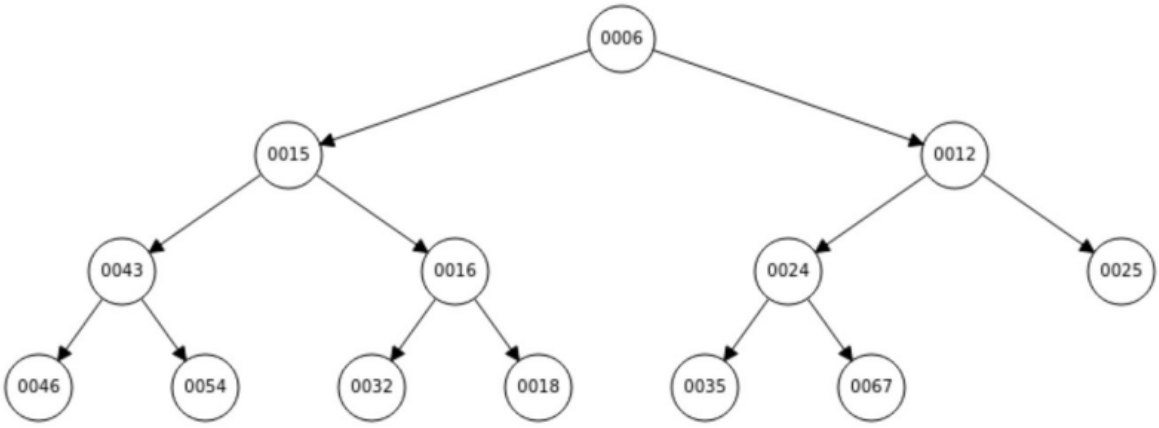
ESERCIZIO D'ESAME HEAP BINARIO

Domanda #3



In uno Heap Binario di tipo "min" l**a 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(5, "elem", A);
```

La chiamata `insert(5, "elem", A)` rappresenta il caso peggiore della operazione insert sullo heap **A**, rispetto alla complessità temporale? Motivare la risposta.

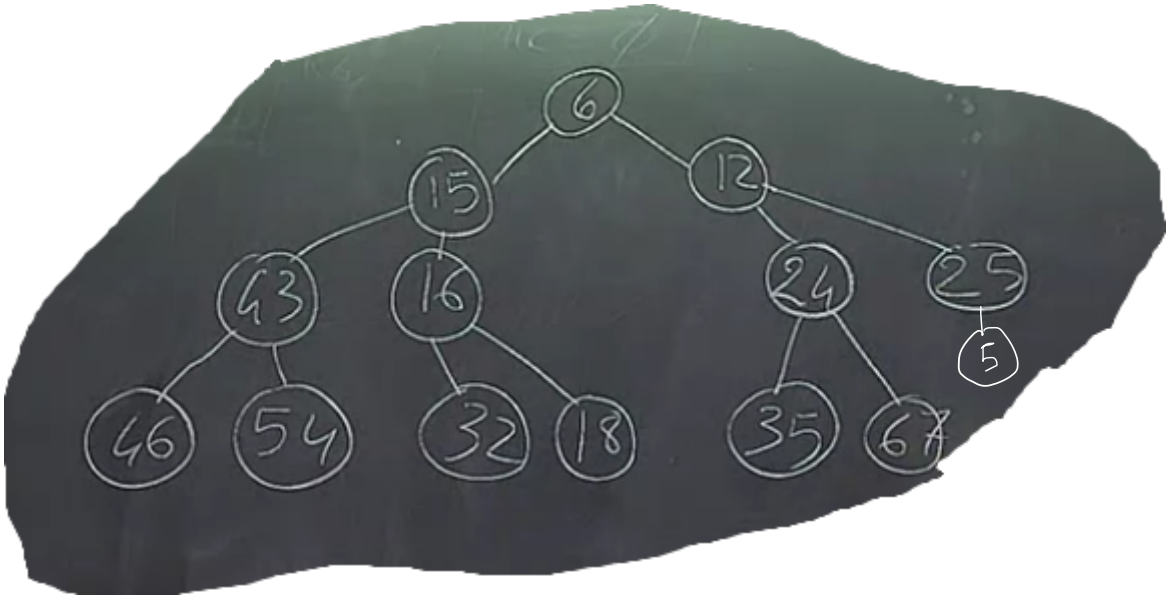
[2/3 del punteggio] Si spieghino dettagliatamente, mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata

```
deleteMin(A);
```

effettuata sullo heap **A** modificato a seguito dell'inserimento dell'elemento con chiave `5`.

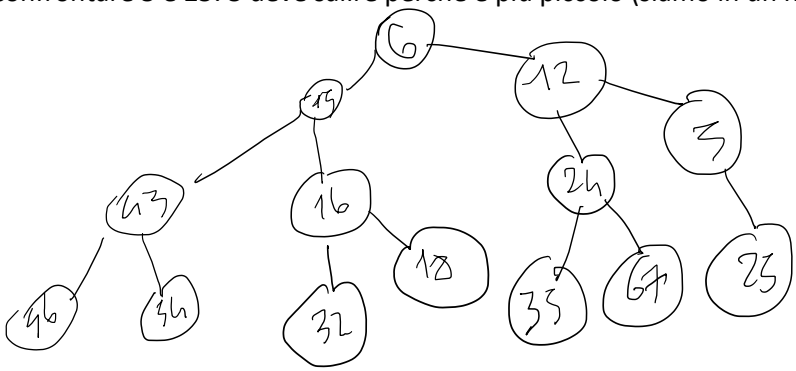
La chiamata `deleteMin(A)` rappresenta il caso peggiore della operazione deleteMin sullo heap A, rispetto alla complessità temporale? Motivare la risposta.

QUELLO CHE DOBBIAMO FARE ALL'INIZIO SULL'ALBERO: inserire il 5 nell' "ultima" posizione disponibile, così da rimanere a una situazione di "albero quasi completo"



DOBBIAMO INSERIRE 5

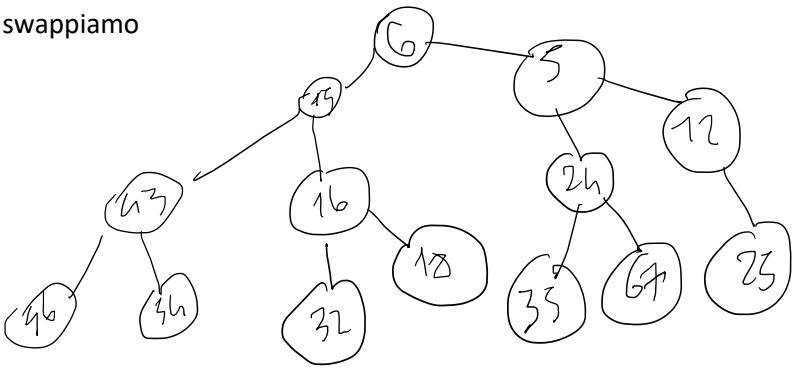
Devo confrontare 5 e 25: 5 deve salire perché è più piccolo (siamo in un heap min)



Swappiamo,

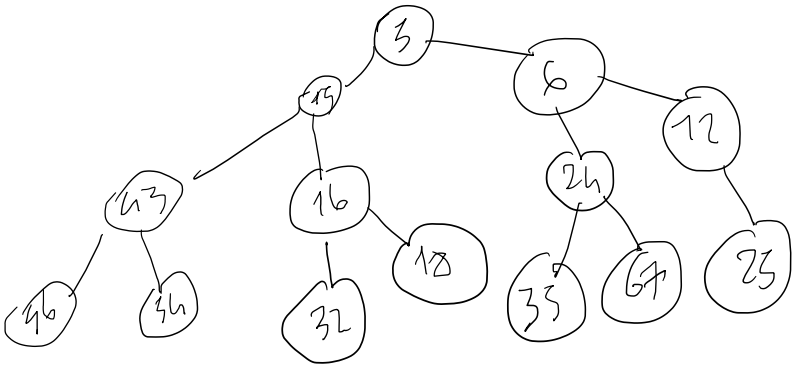
confrontiamo 5 con 12, 5 è più piccolo, deve salire

swappiamo



confrontiamo 5 con la radice 6, 5 è più piccolo, deve salire,

swappiamo,



siamo arrivati a mettere 5 nel posto giusto.

Abbiamo fatto tanti scambi quanto è l'altezza dell'albero, in uno heap binario l'altezza dell'albero è $\log_2 n$

Quindi la complessità di questo inserimento (con spostamenti) è in theta di log n. (n è numero di nodi).

Abbiamo risposto alla prima domanda.

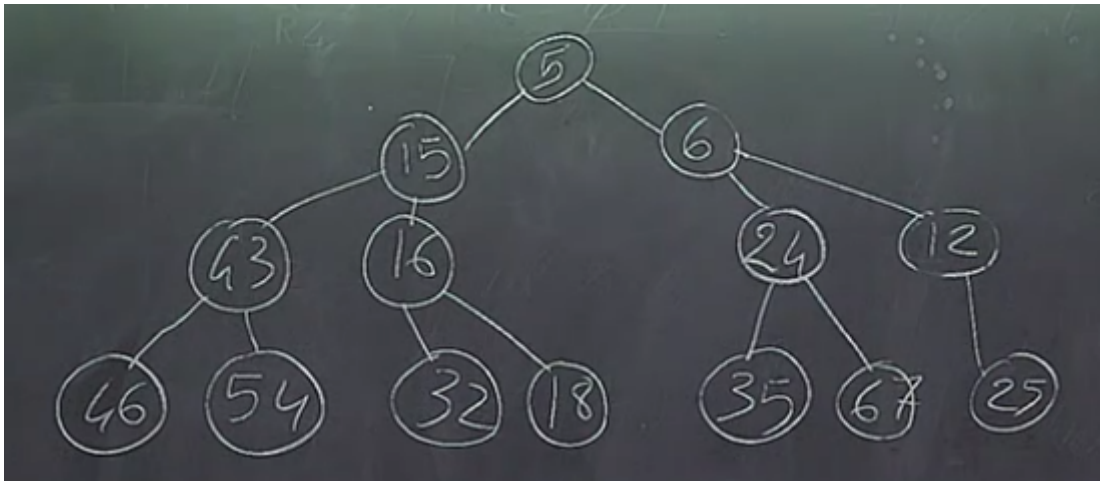
[2/3 del punteggio] Si spieghino dettagliatamente, mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata

```
deleteMin(A);
```

effettuata sullo heap **A** modificato a seguito dell'inserimento dell'elemento con chiave `5`.

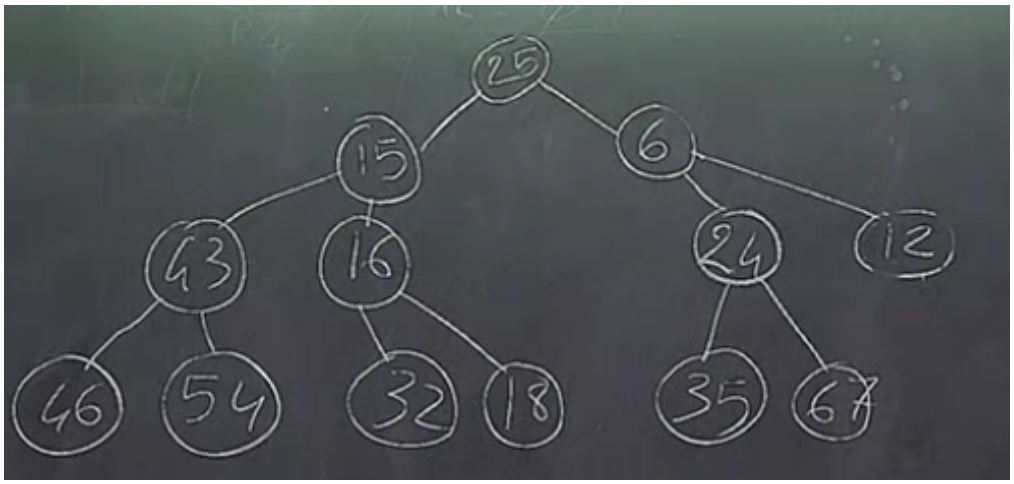
La chiamata `deleteMin(A)` rappresenta il caso peggiore della operazione deleteMin sullo heap A, rispetto alla complessità temporale? Motivare la risposta.

Lo heap con 5 inserito nel posto giusto è questo:



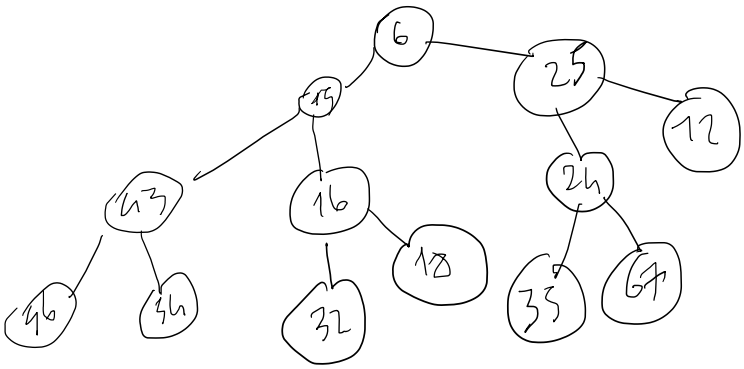
Noi vogliamo togliere il contenuto della radice. (il nodo con chiave minima si trova alla radice e noi dobbiamo fare deleteMin)

Quindi facciamo così



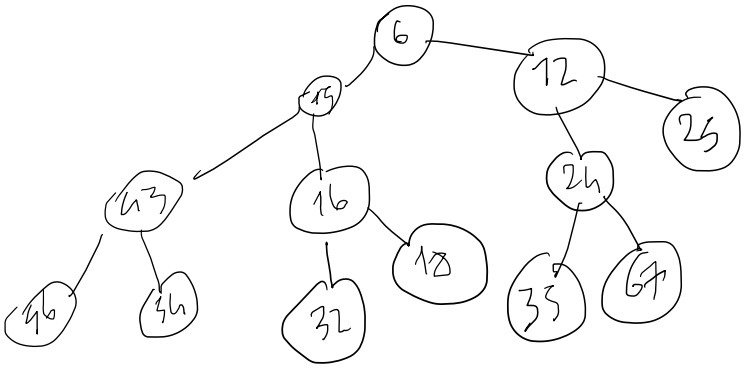
Così facendo non abbiamo distrutto le proprietà strutturali, però il 25 non è nel posto giusto.

Confronto 25 con 6 e con 15, e vado a sostituirlo con quello dei suoi figli che ha chiave minima, lo swappo quindi con 6.



Ora devo confrontare il 25 con i suoi figli 12 e 24.

Swappo quindi il 25 con quello dei suoi figli che ha chiave minima, quindi con il 12.



Sono arrivato a mettere 25 come foglia (non ha più figli). Ho finito.

Non siamo nel caso peggiore, ma quasi. (nel caso peggiore in assoluto sarebbe se avessimo fatto un altro swap, quindi se quel nodo avesse avuto un altro/due altri figli. Quindi fossi arrivato all'ultimo livello, dal primo che è la radice, quindi sarei in theta(log n).