

Domanda #2

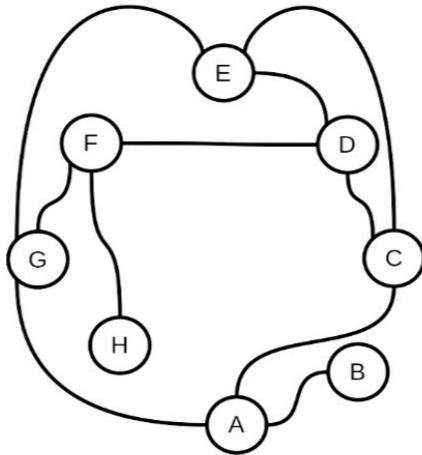
Si consideri il grafo non orientato e non pesato disegnato sotto:

1

[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 **B** con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

2

[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 **C** con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.



Esercizio 1

1) Partendo da B posso visitare tutti i nodi?

A:



2) Da A posso visitare tutti i nodi?
G e C, posso visitare C:



3) DA C POSSO VISITARE D O E
DEPOIS DA VISITARE D:



4) DA D POSSO VISITARE E ED F, DEPOIS
DA VISITARE E:

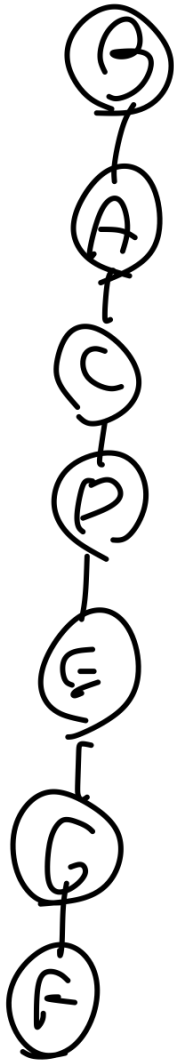


5) DA E POSSO ANDAR, SOLO DA G:





6) DA G VISITA F VISITA CHE A ED F
 sono GRA MANCANTI:



7) INTRINSECA A POSSO ANALIZZARE solo
 DA H:



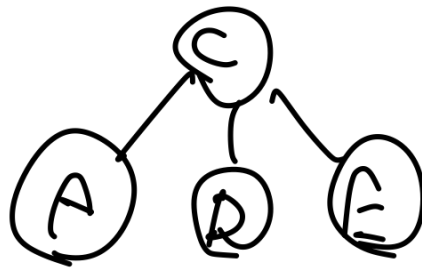
N.B. QUESTA **NON** È L'UNICA
 SOLUZIONE, CI SONO ALTRE PENSIERI
 DA POTER VEDERE, AD ESEMPIO:



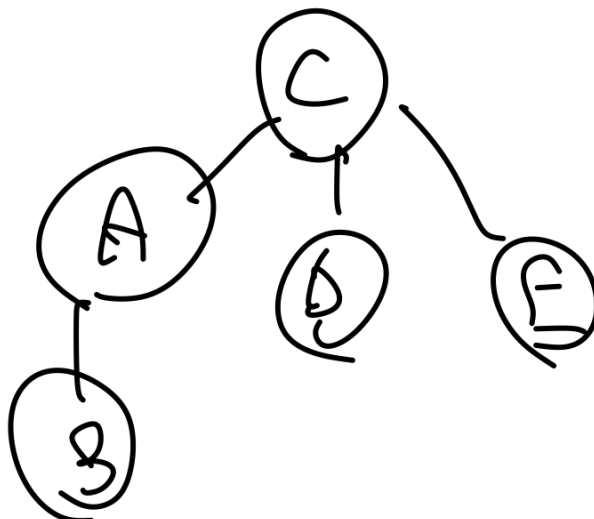


2) DA C POSSO VISITARE JOINTS

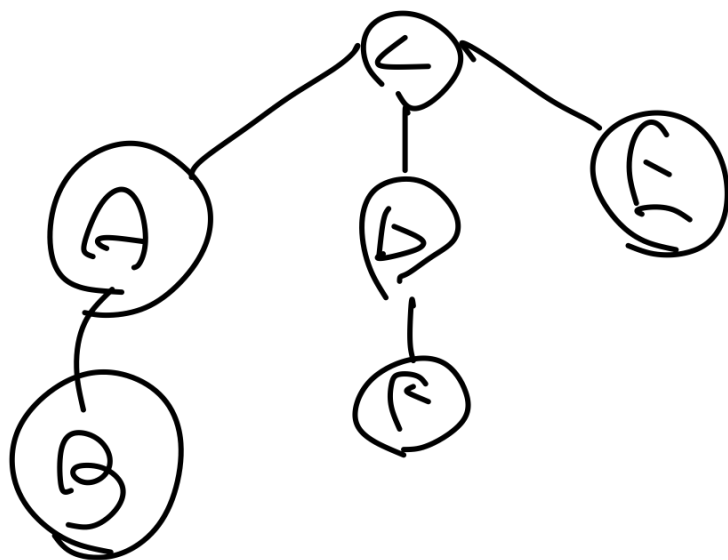
A, D E F QUINDI:



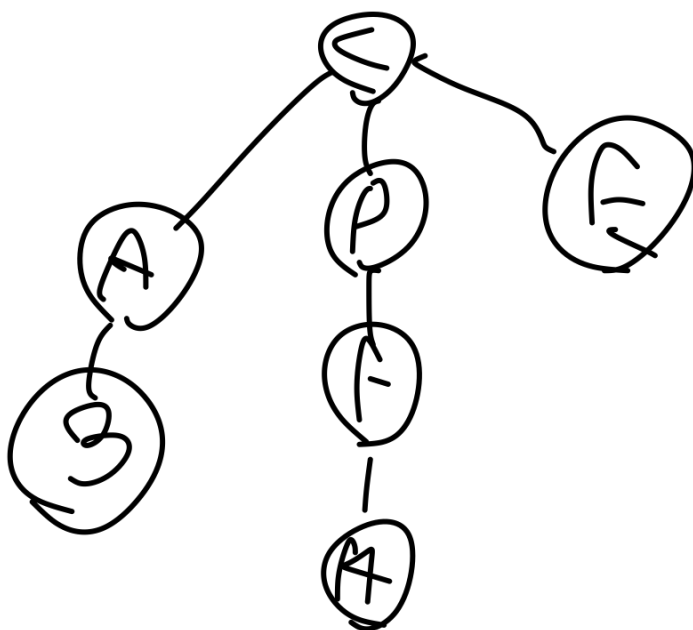
A lo posso collegare a B:



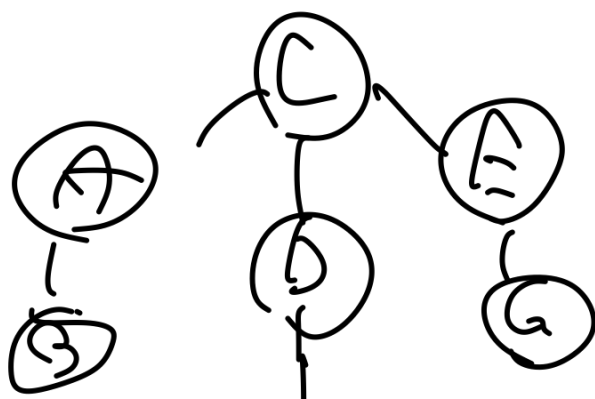
GA is Polish notation: F



DA F is Polish notation: H



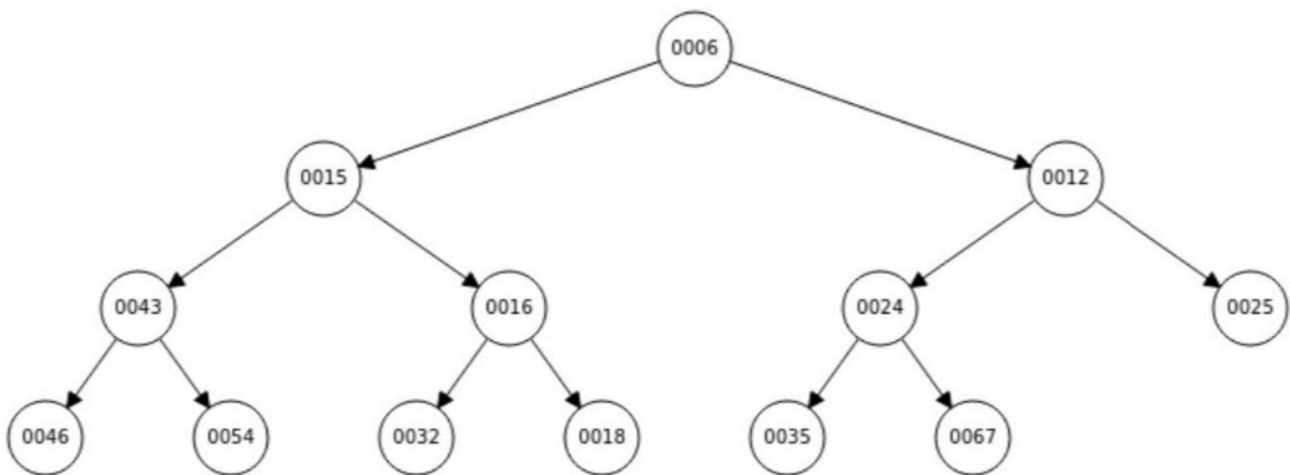
INF is: G A G





Anche qua abbiamo invece
 l'ordinamento, ma il funzionamento è
 analogo.

Domanda #3



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

[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

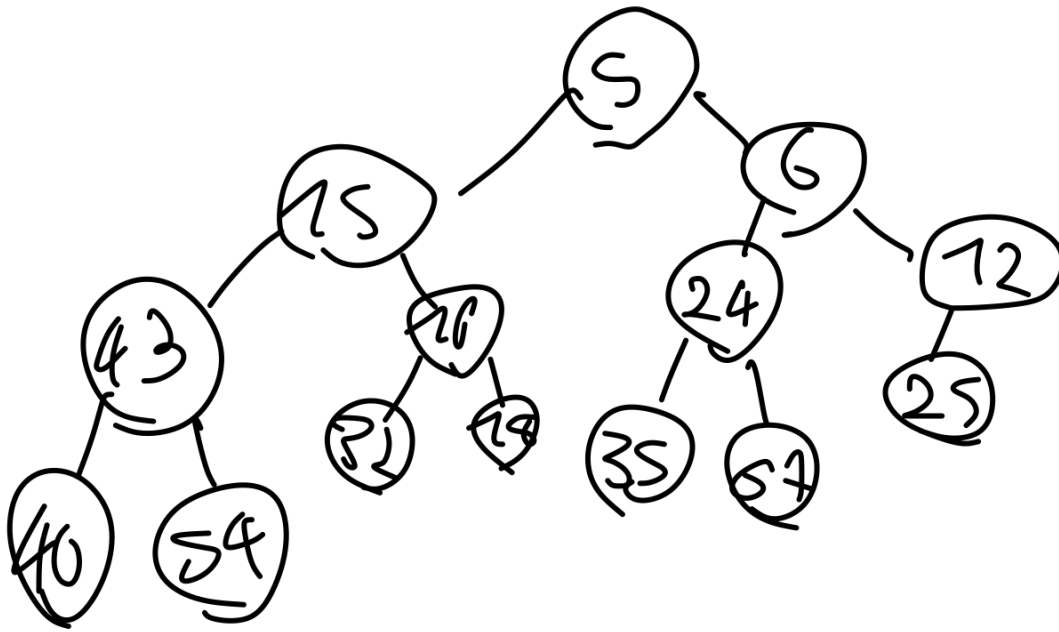
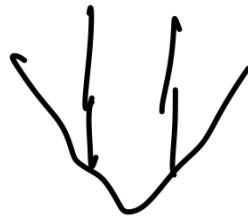
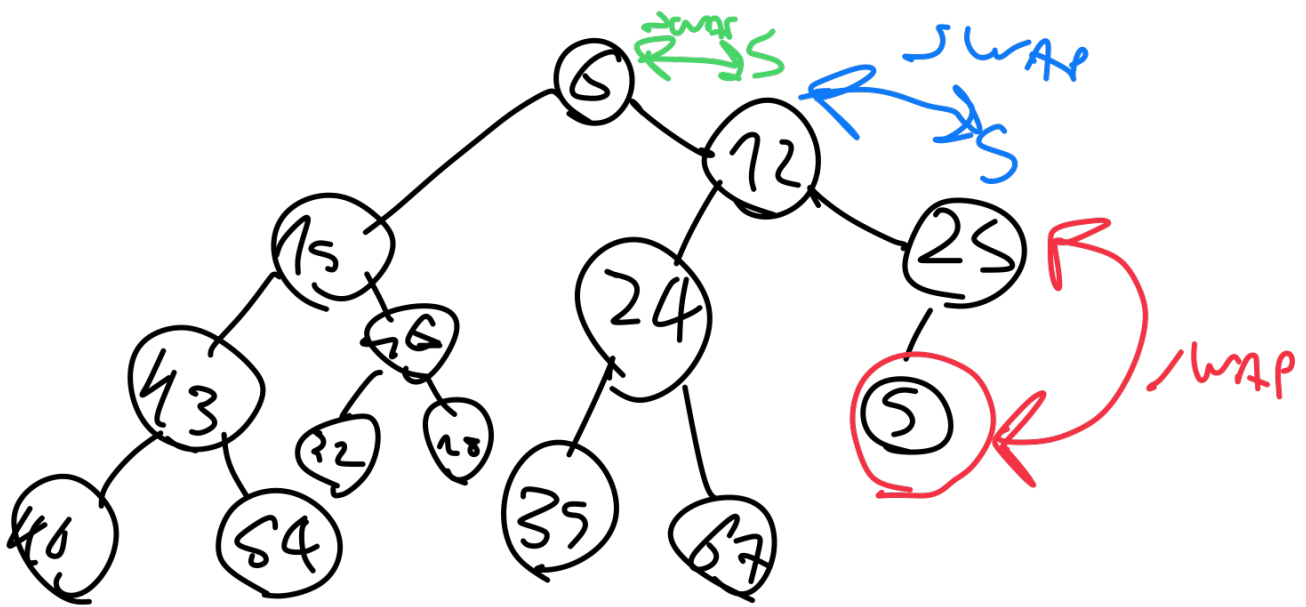
[2/3 del punteggio] Si spieghino dettagliatamente, mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata

```
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.

1)

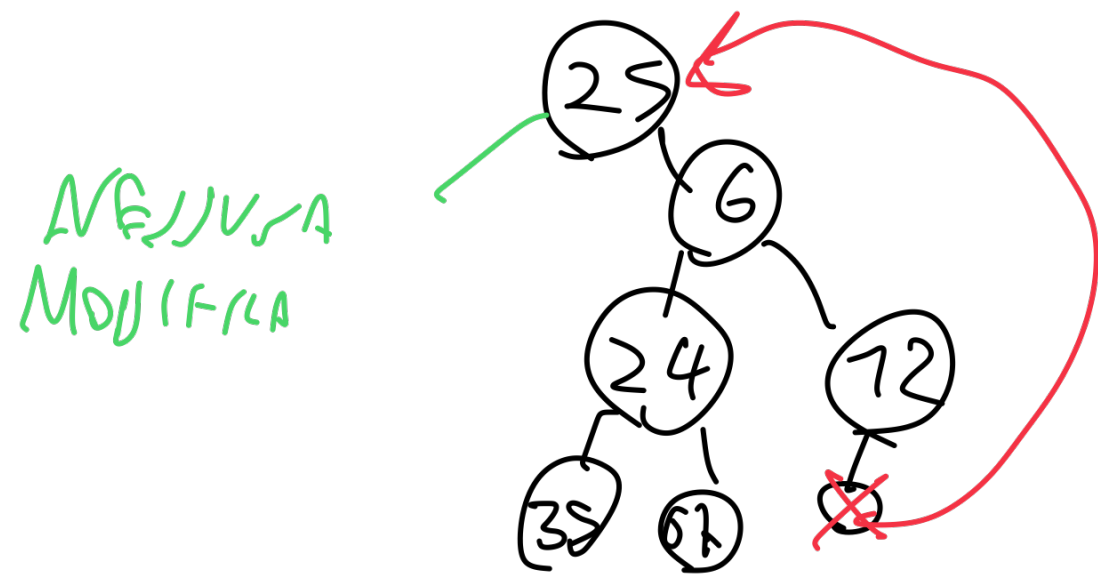


VRSTO CME DOBBAVAMO VRAVINE L'AFER
 AKAVRANO VRAI CAVO PEFERONIC: $\Theta(\log m)$

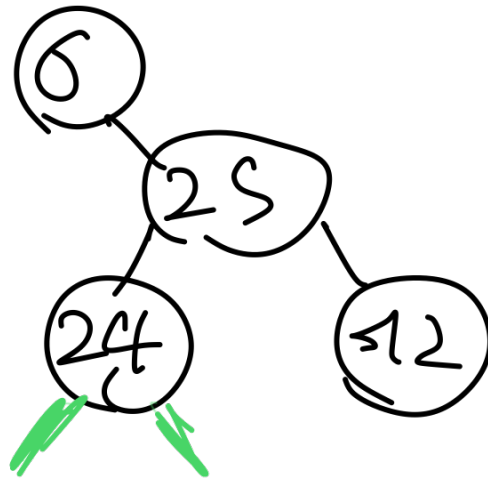
2) LA FUNZIONE DELETE ELIMINA LA
 NODICE SOSTITUENDO IL SUO VALORE CON
 IL NODO A PIU' DESTRA ALL'INTERNO DELLA
 ALBERO.

Questa operazione però compromette le
 proprietà dell'albero, per questo motivo si
 utilizza una funzione, AUXILIARE CHE
 MI AIUTA A RIPRISTINARE QUELLE
 PROPRIETA', LA FUNZIONE IN QUESTO CASO E'
 CHIAMATA MODIBALZO().

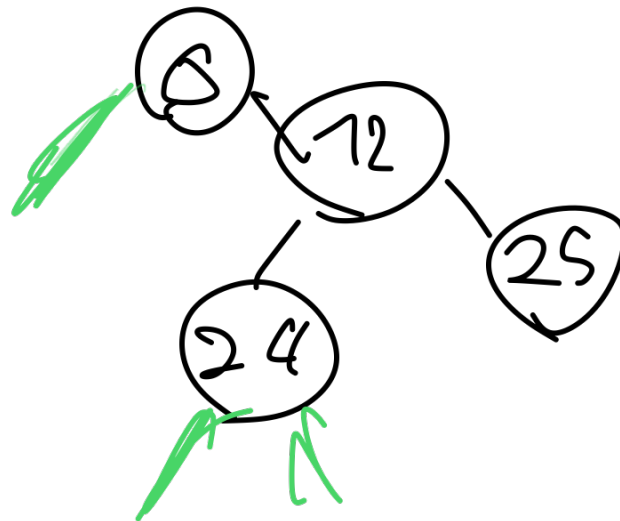
NEL NOSTRO ALBERO ANDRANNO AD ELIMINARE
 IL 25 E SWAPPANDO.



IMPLEMENTING A FEW 2100s MOV BAZS
IN QWAND 2S F' PRV' CRAWL. DI 6:



SWAPPING 2S COL 12:



ADDITIONAL LIMITS, COMPLEXITY ($\log n$)