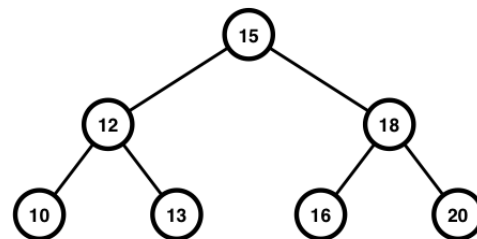


ESERCIZIO 1

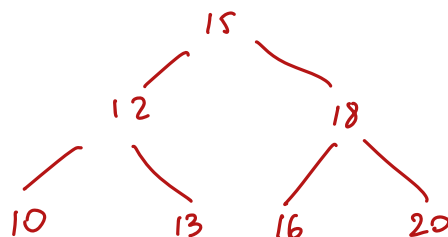
- (a) Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up.

Quindi si eseguano nell'ordine dato le seguenti operazioni sullo splay tree a lato:

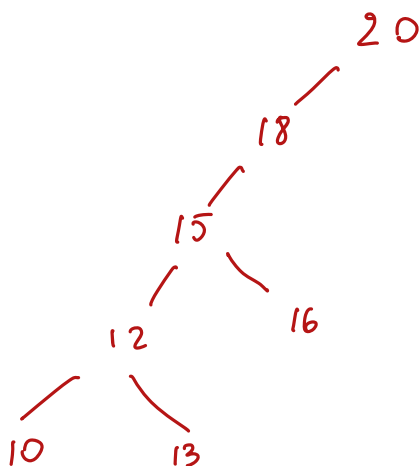
- SEARCH 20, 13
- INSERT 17
- DELETE 15



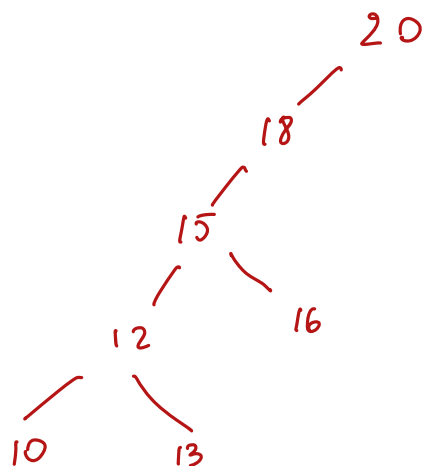
- (b) Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig*, nonché l'operazione di assemblaggio finale, in un splay tree di tipo top-down.

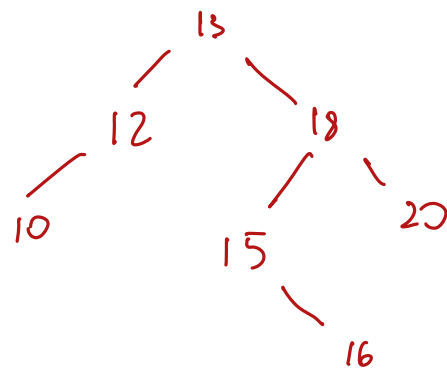
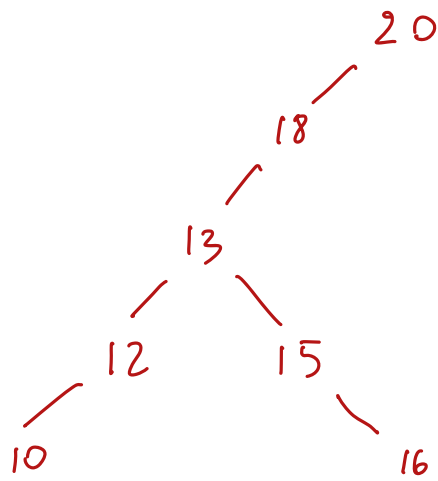


SEARCH (20)

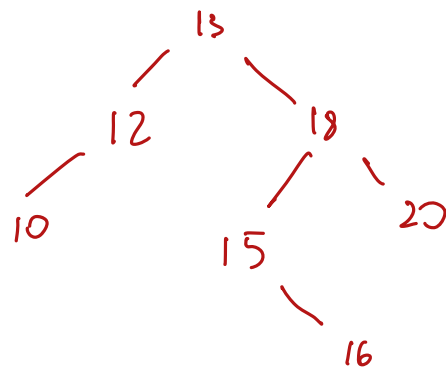


SEARCH (13)

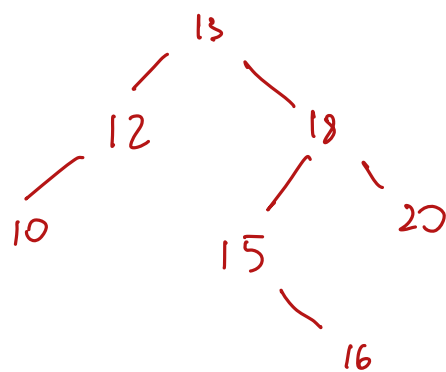


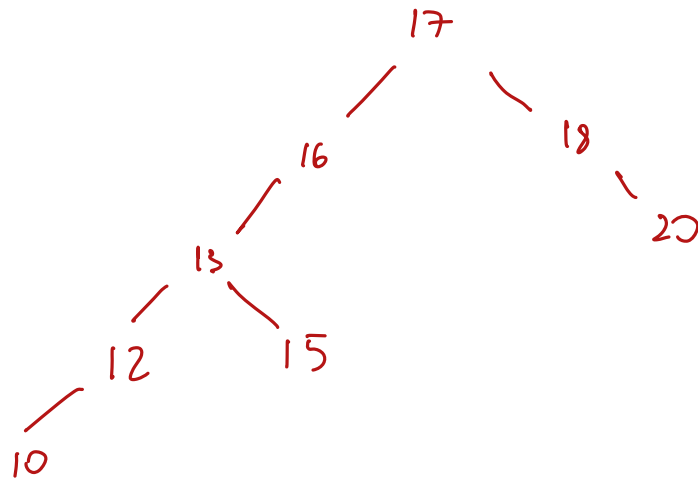
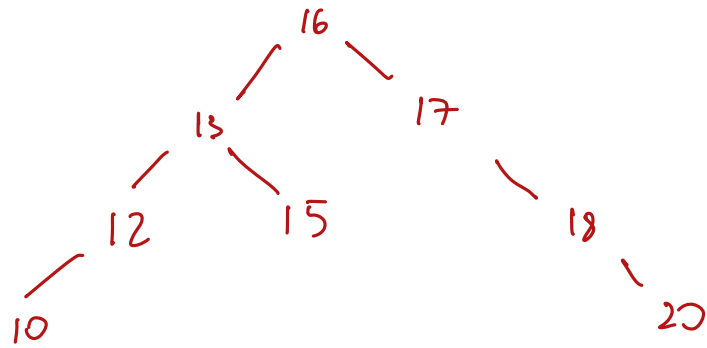
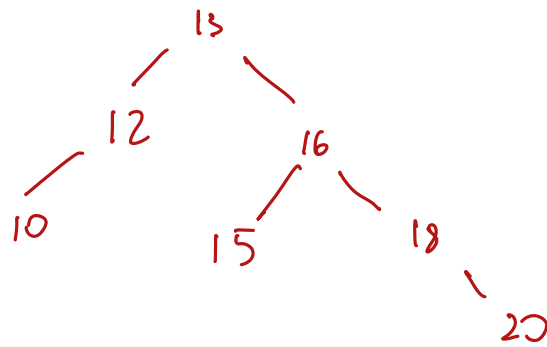


INSERT (17)

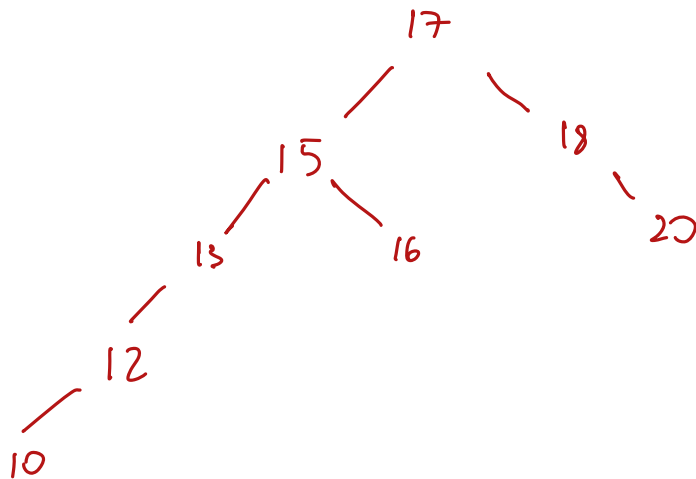


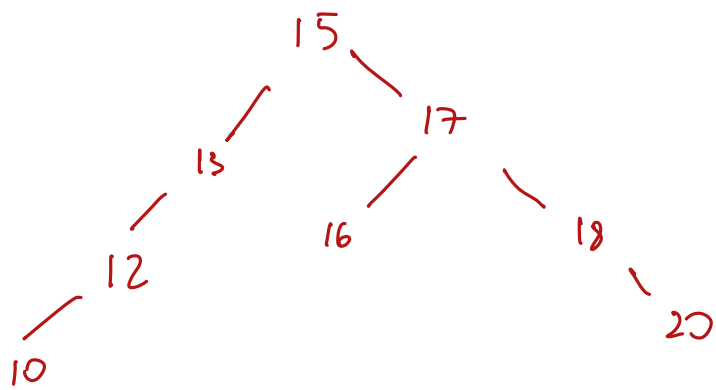
(SPLAY (17))



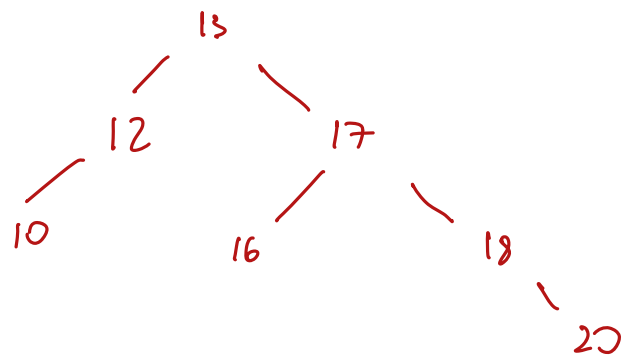


DELETE (15)





15



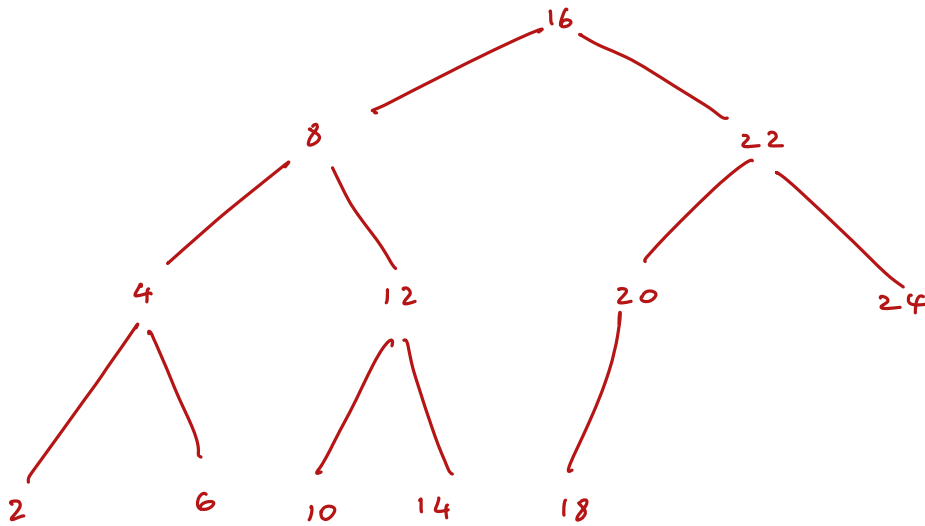
ESERCIZIO 3 (Splay trees)

Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up.

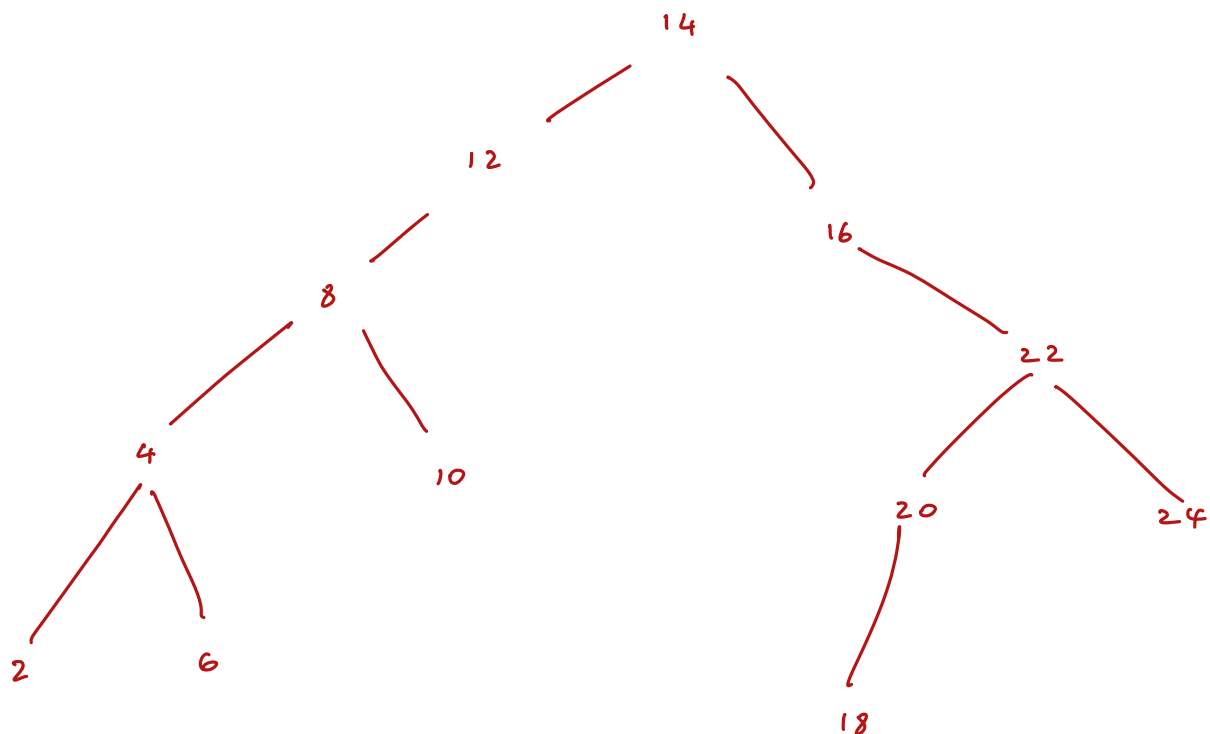
Quindi si eseguano nell'ordine dato le seguenti operazioni su uno splay tree la cui configurazione iniziale è quella di un albero binario completo contenente le 12 chiavi $\{2i : 1 \leq i \leq 12\}$:

- SEARCH 14, 2, 24
- INSERT 17
- DELETE 22
- SEARCH 2

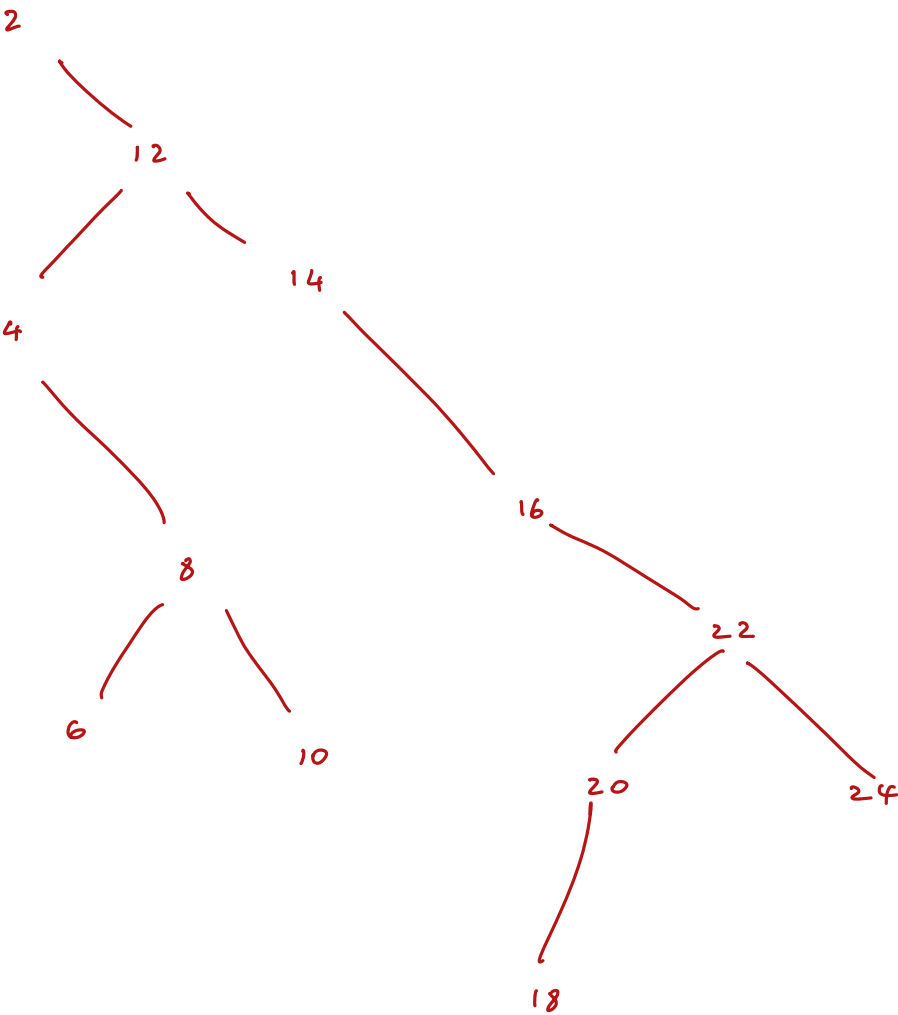
Nota bene: Si ricorda che un albero binario si dice *completo* quando tutti i suoi livelli, con al più l'eccezione dell'ultimo, sono completi e tutti i nodi nell'ultimo livello si trovano il più a sinistra possibile.



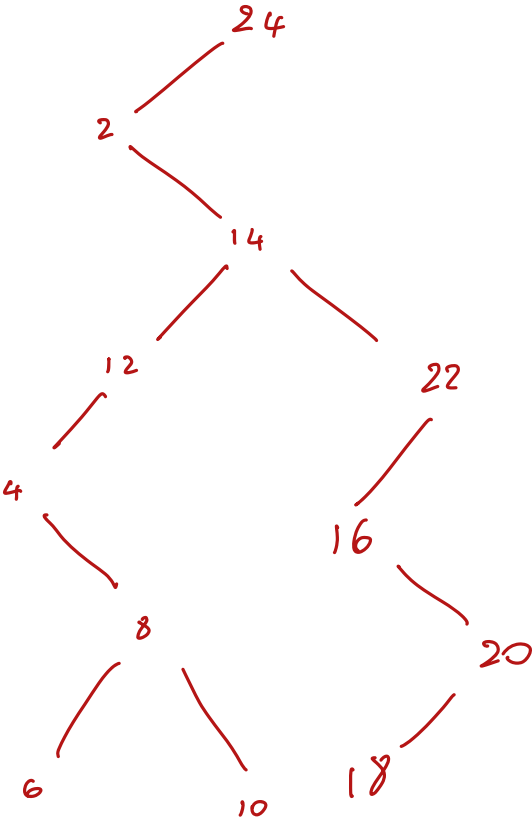
SEARCH (14)



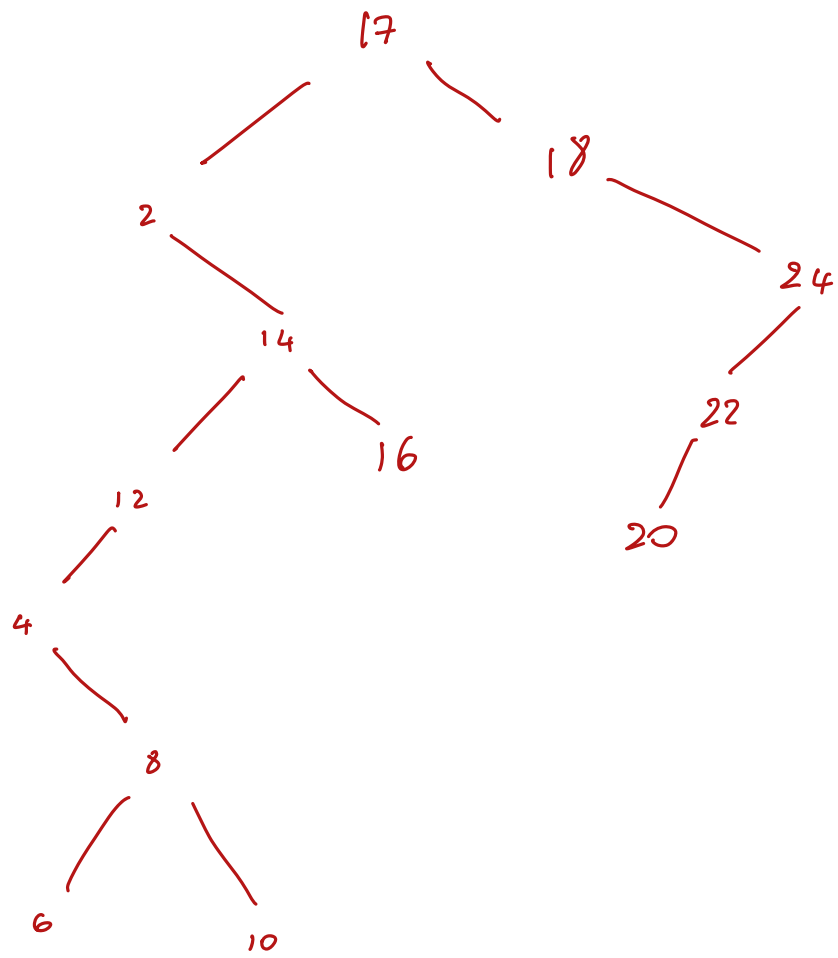
SEARCH (2)



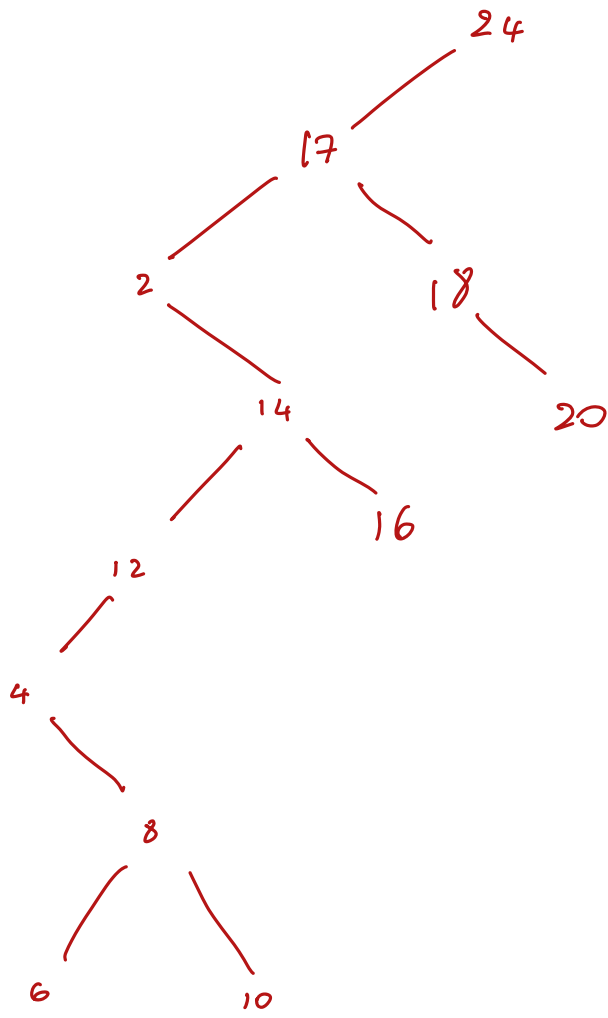
SEARCH (24)



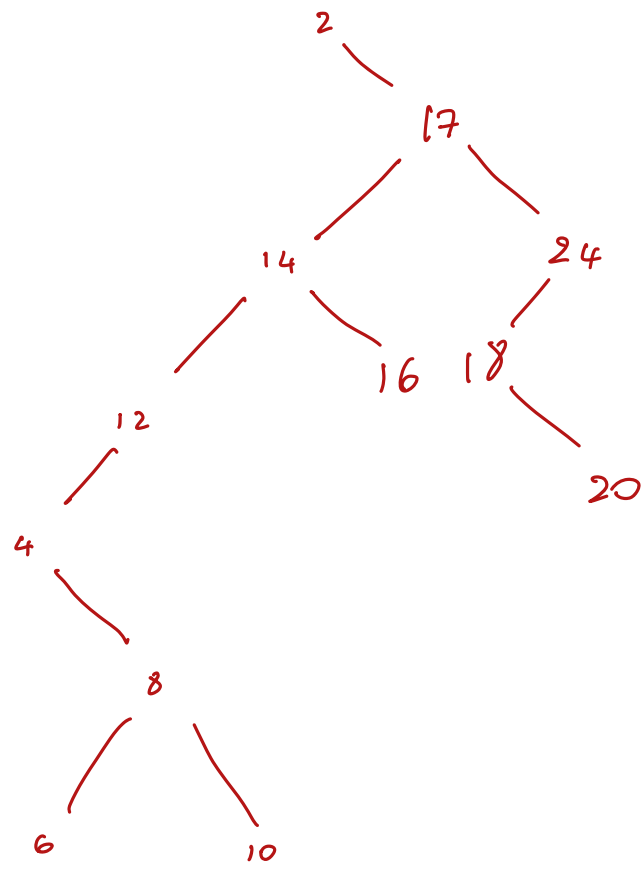
INSERT (17)



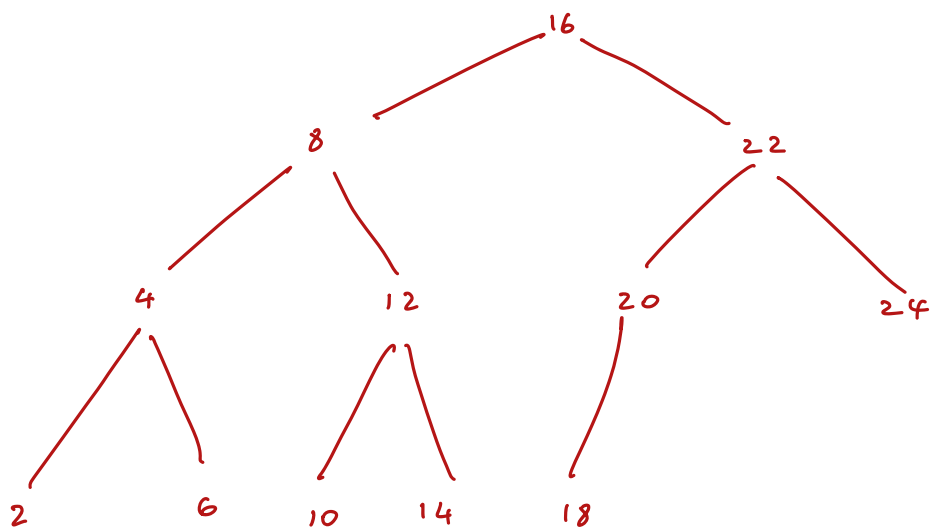
DELETE (22)



SEARCH (2)

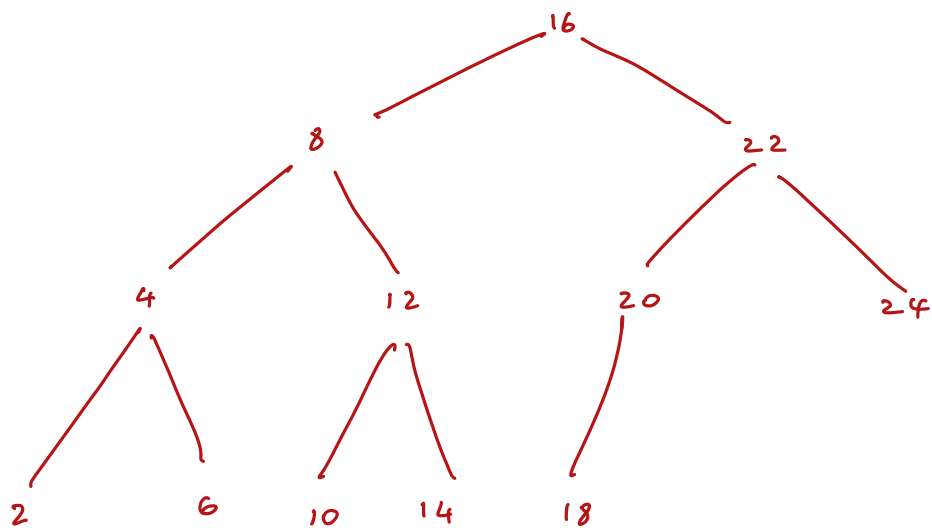


E IN MODALITA' TOP-DOWN ...



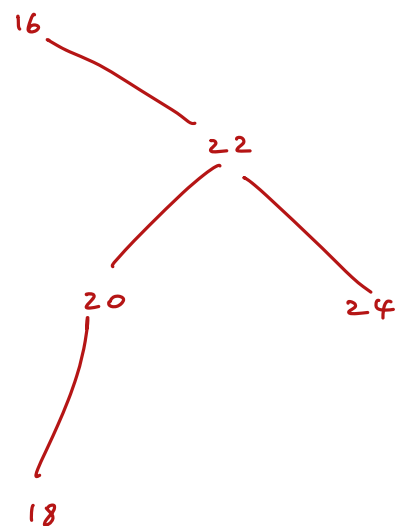
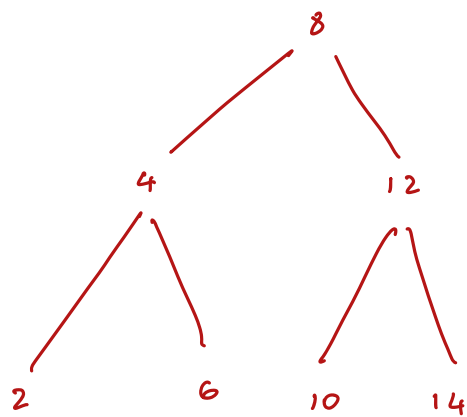
SEARCH (14)

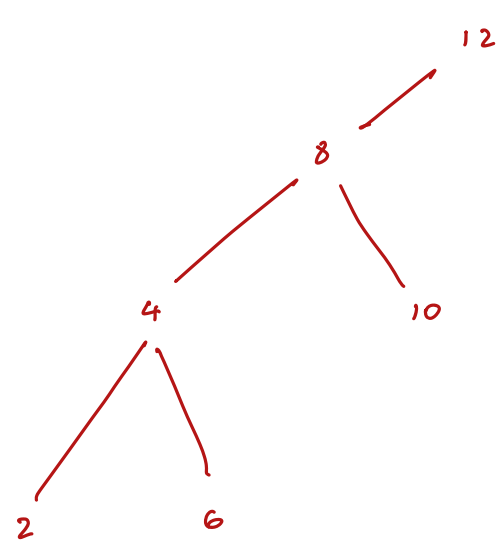
\emptyset



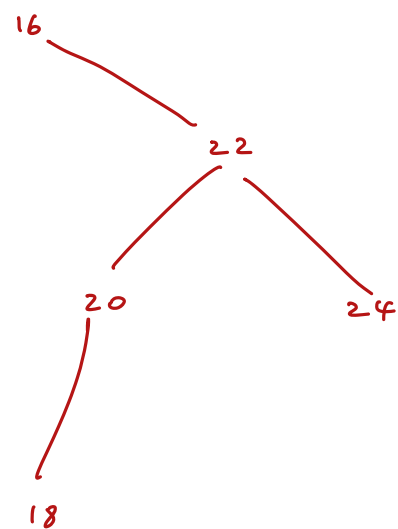
\emptyset

\emptyset

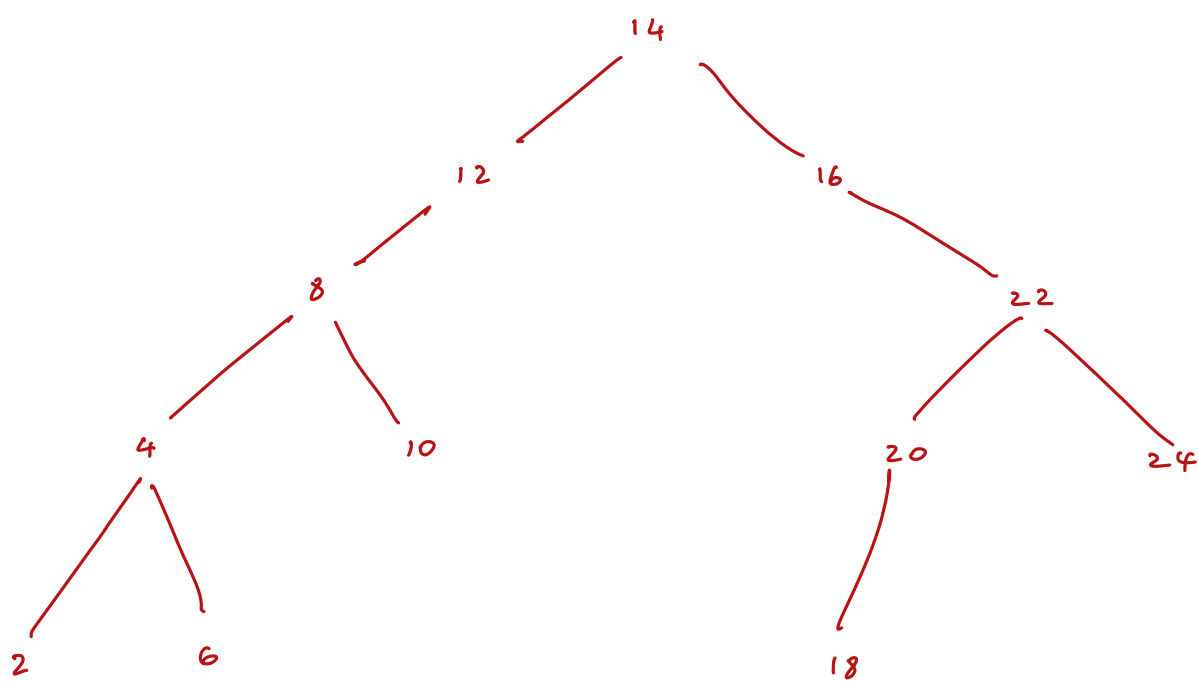




14



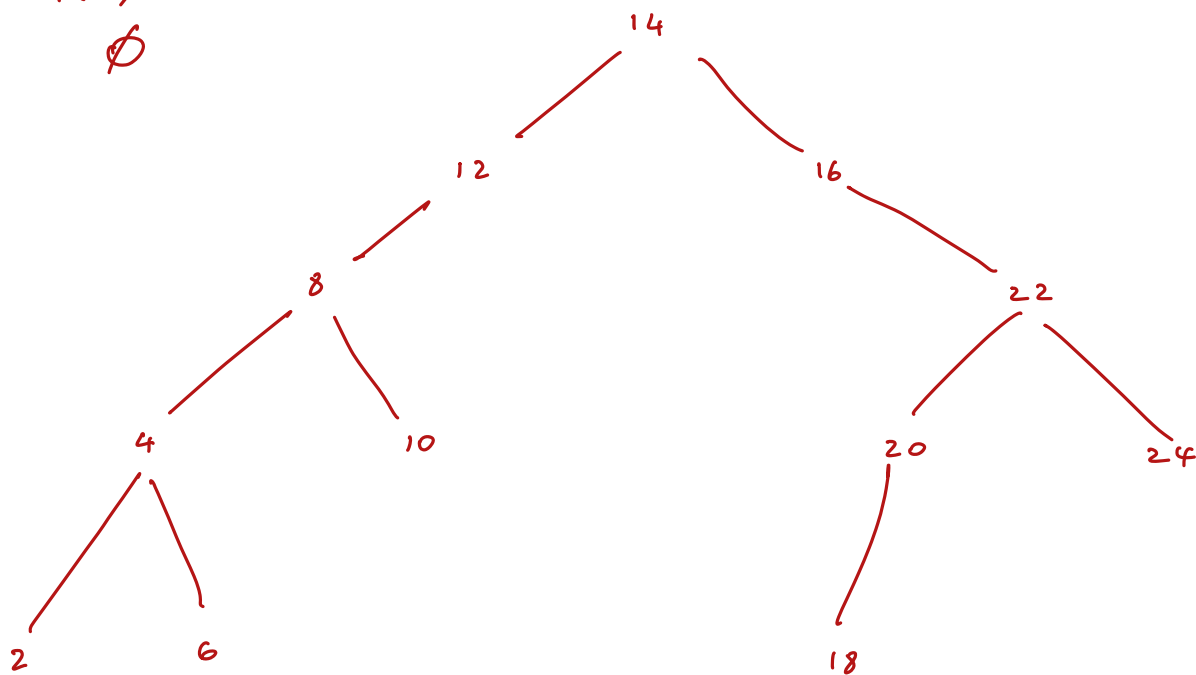
RIASSEMBLAGGIO



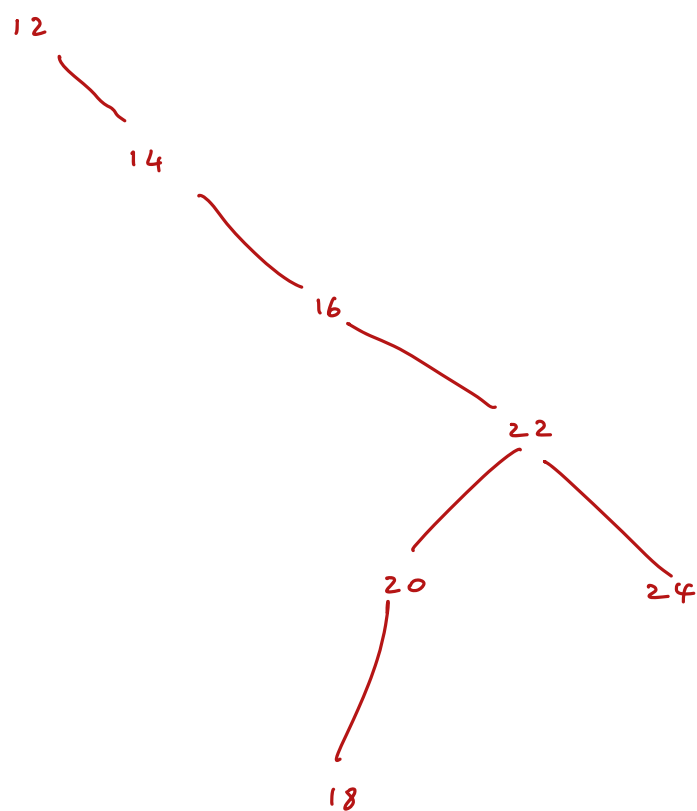
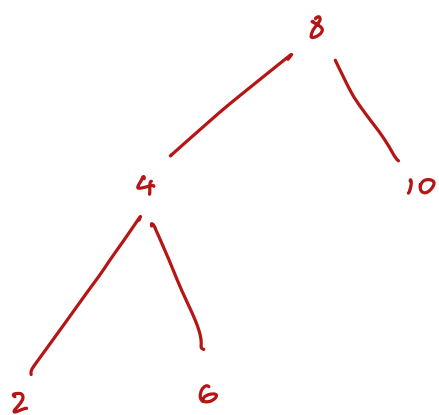
SEARCH(2)

\emptyset

\emptyset

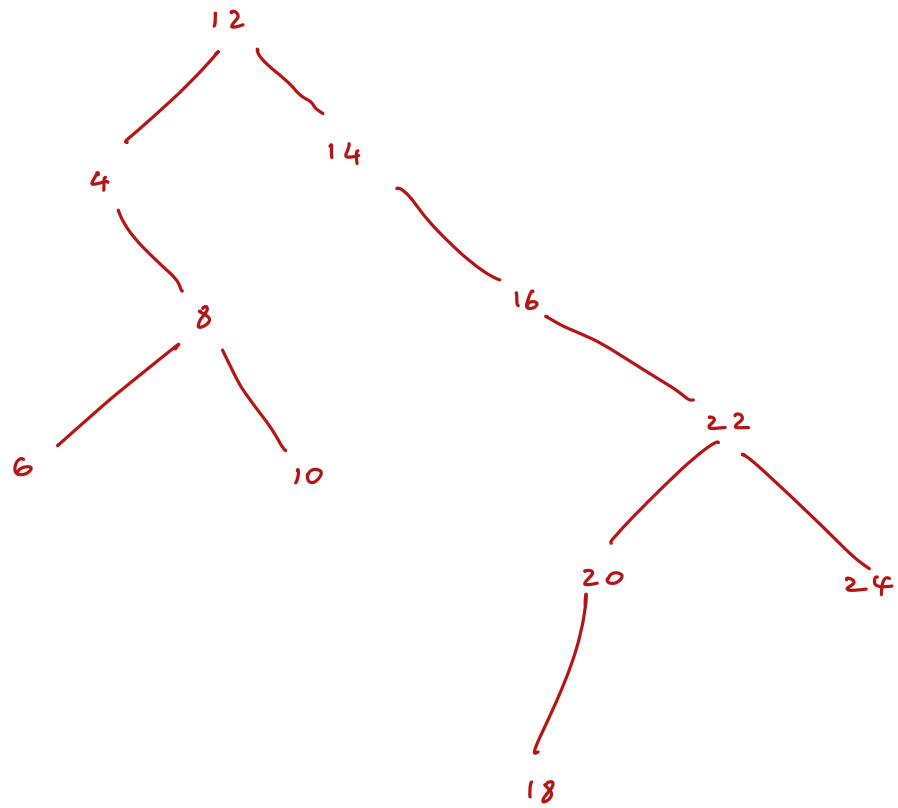


\emptyset

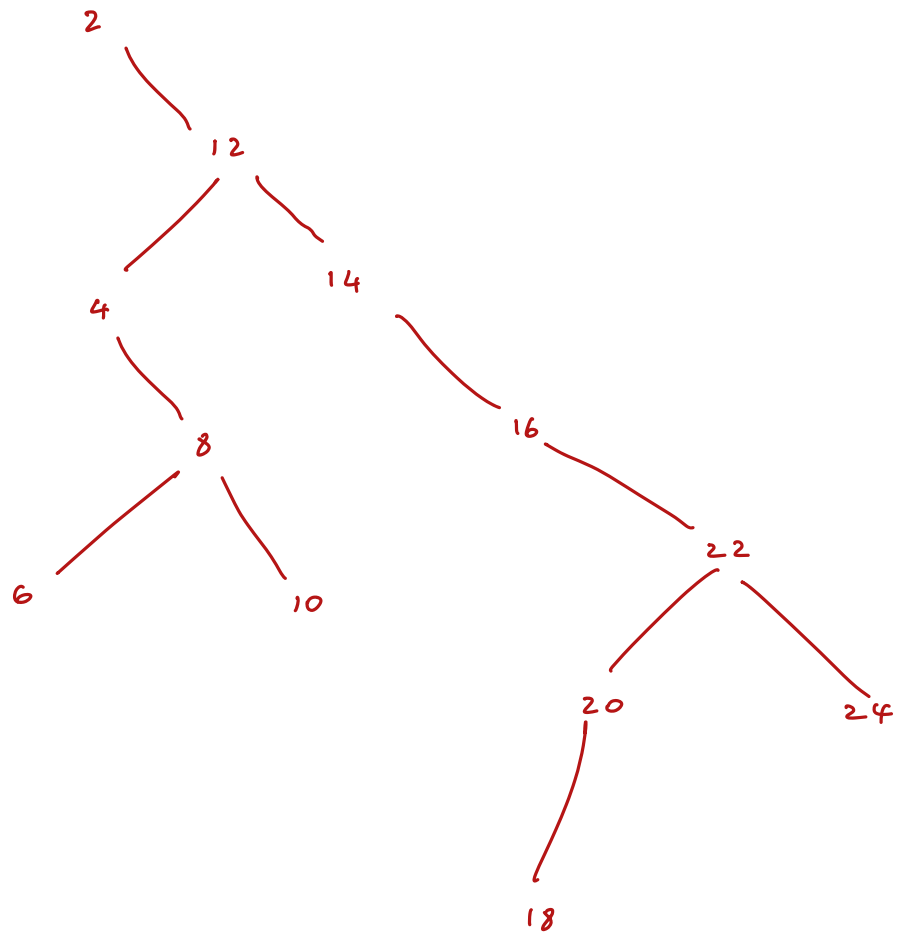


\emptyset

2

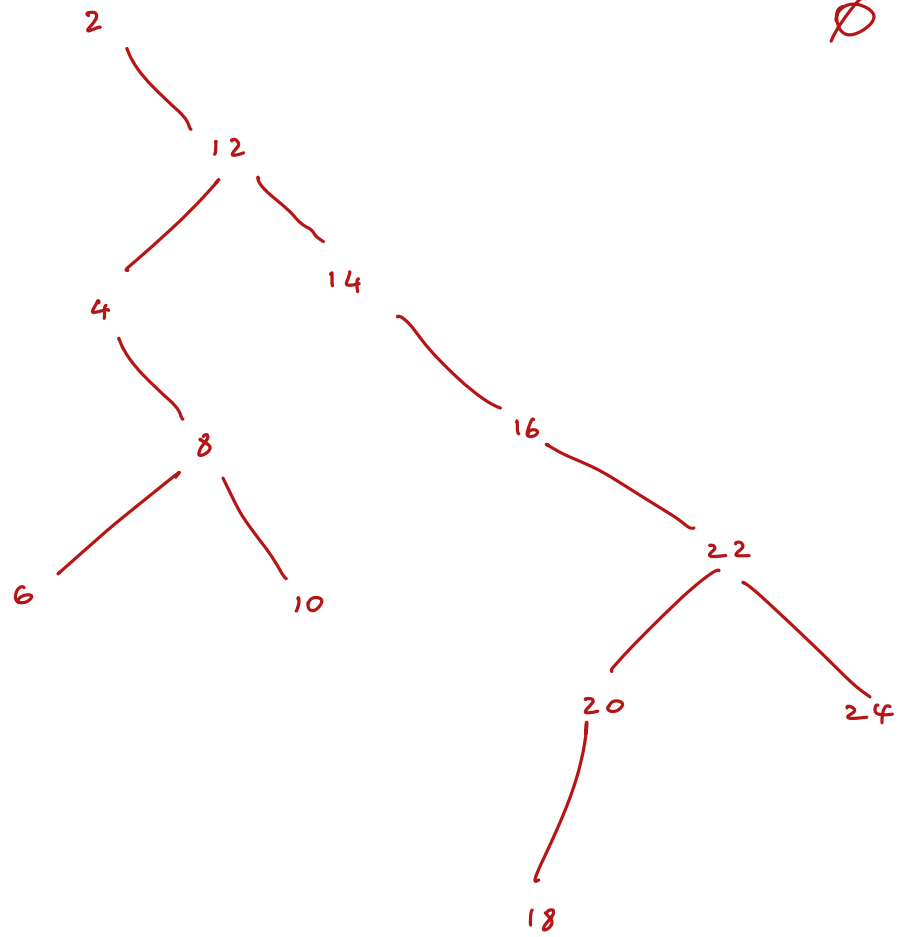


RIASSEMBLAGGIO

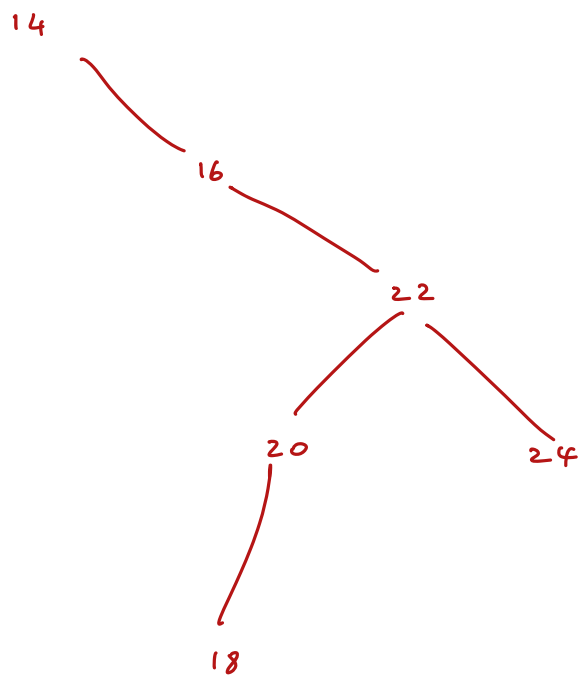
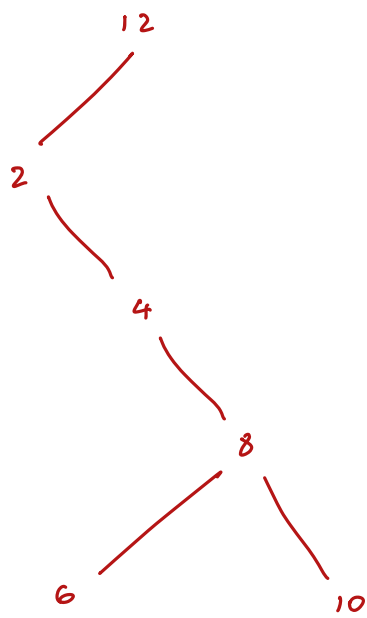


SEARCH (24)

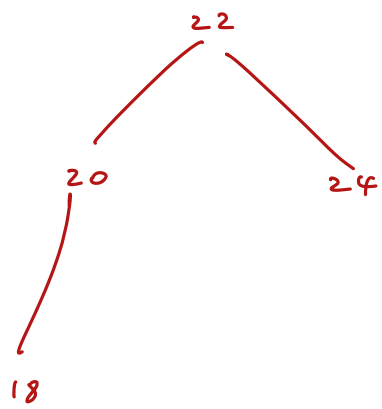
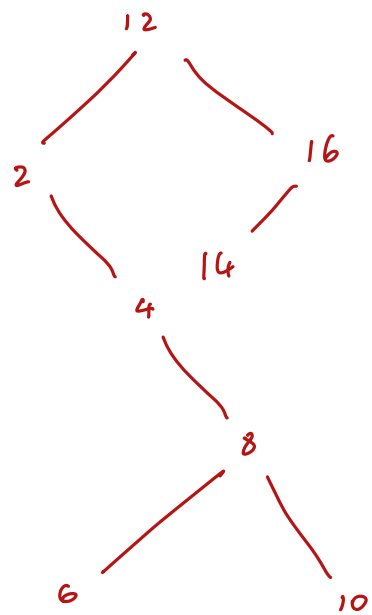
\emptyset



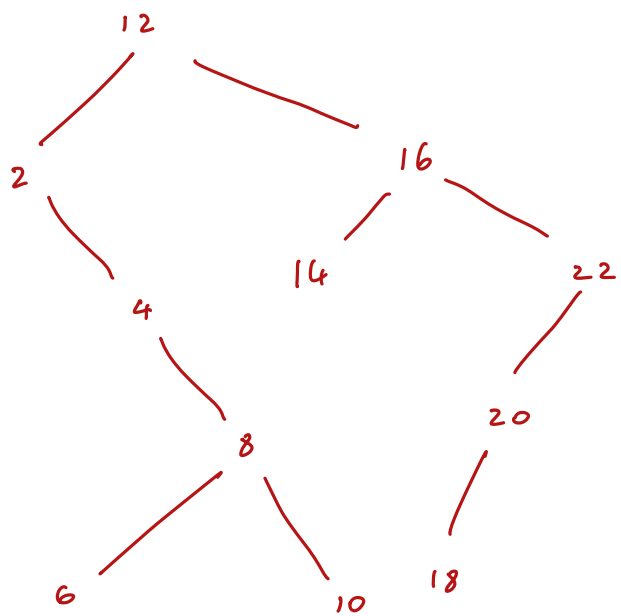
\emptyset



\emptyset



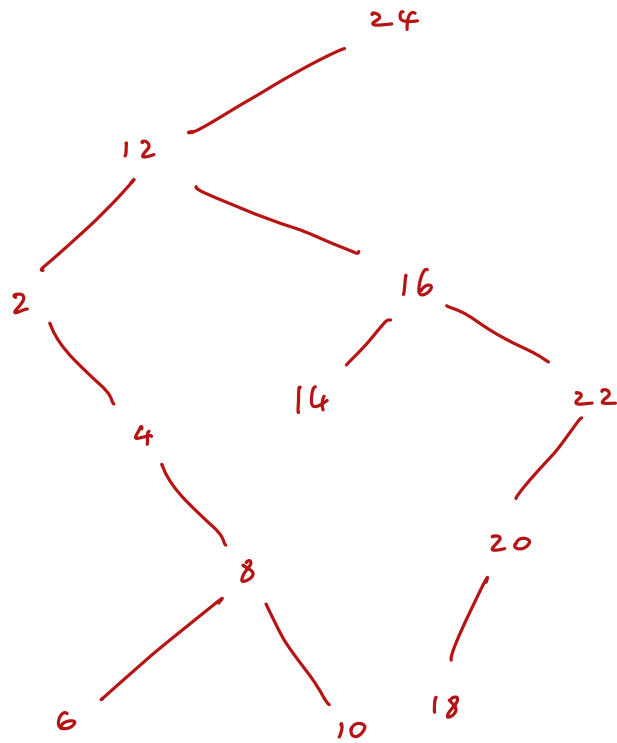
~~Ø~~



24

~~Ø~~

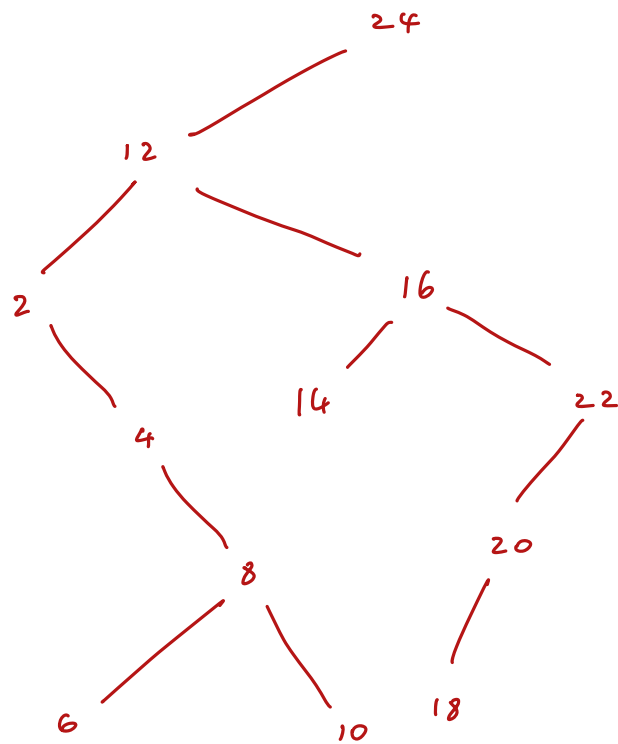
RIASSEMBLAGGIO



- SEARCH 14, 2, 24
- INSERT 17
- DELETE 22
- SEARCH 2

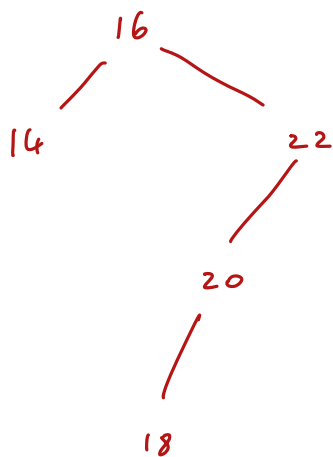
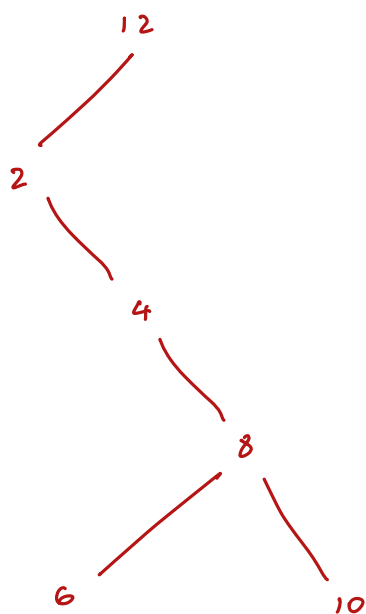
INSERT(17)

Ø

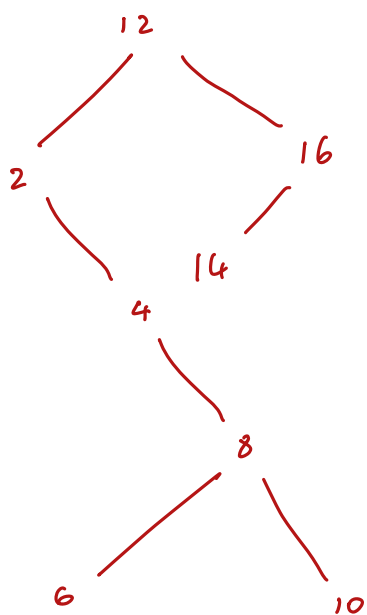
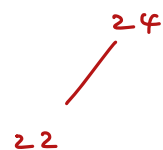
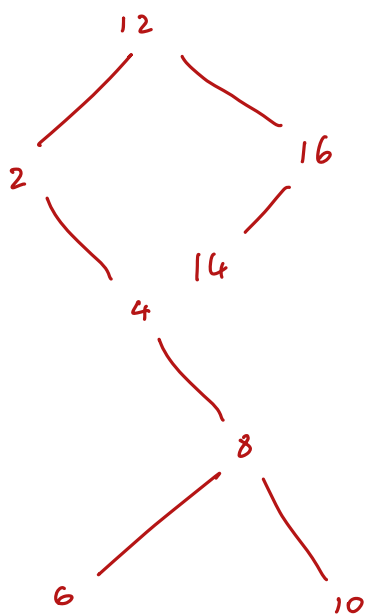


Ø

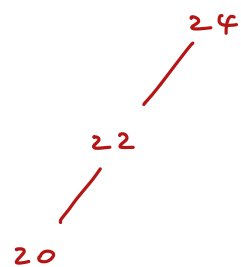
(CON ZIG-ZAG NON SEMPLIFICATA)



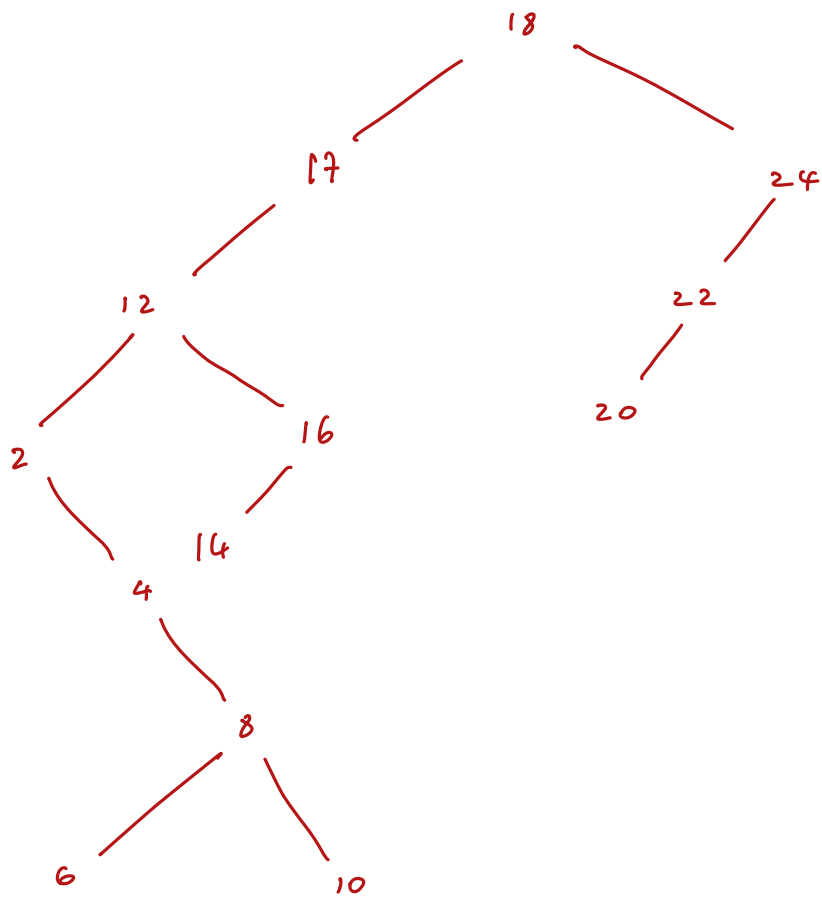
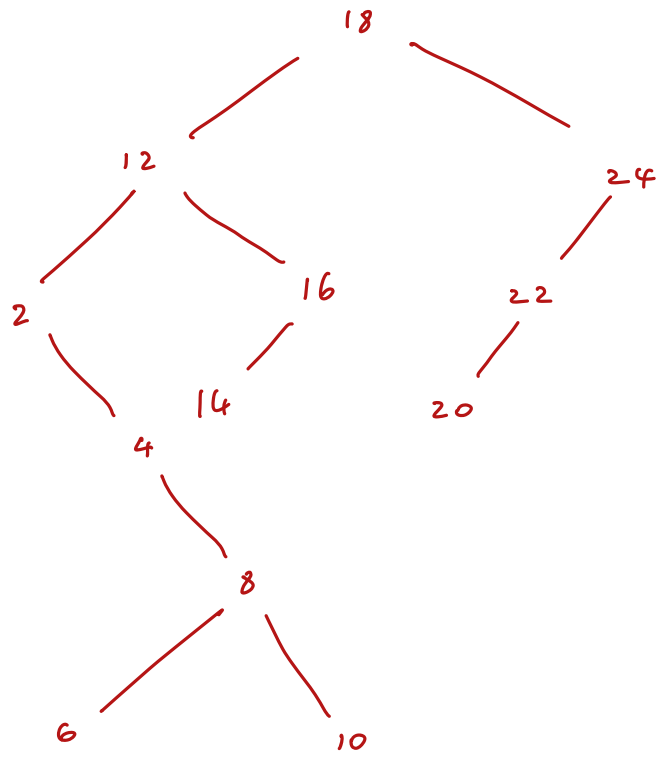
24

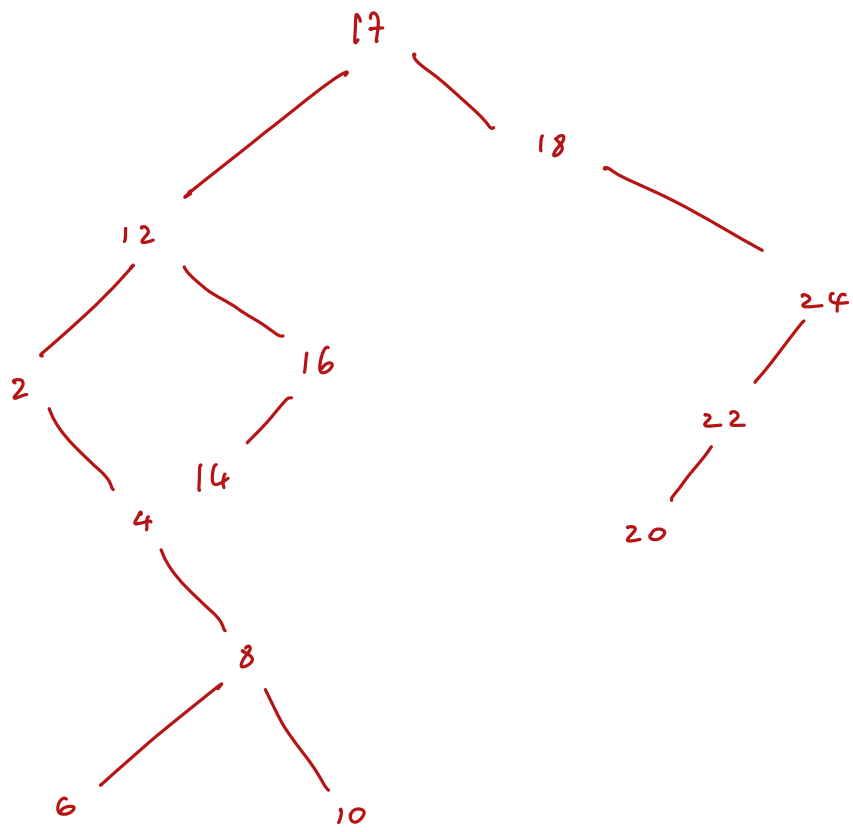


18



RIASSGMBLACCIO FINALE

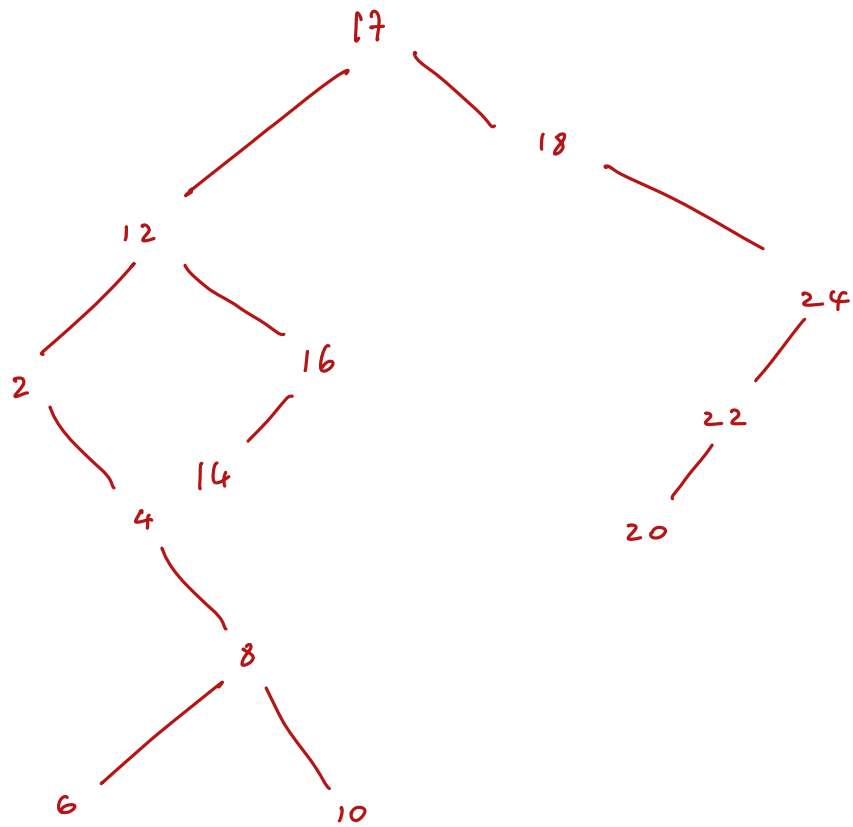


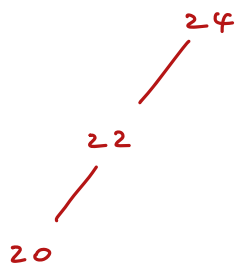
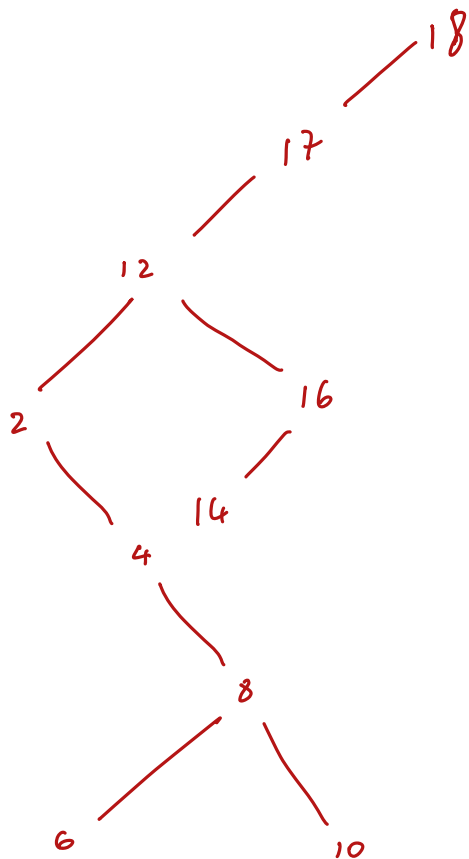


DELETE (22)

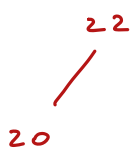
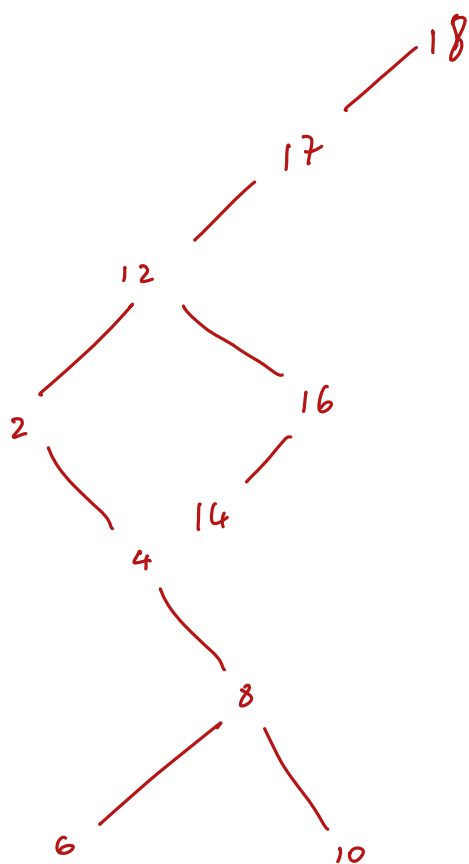
\emptyset

\emptyset



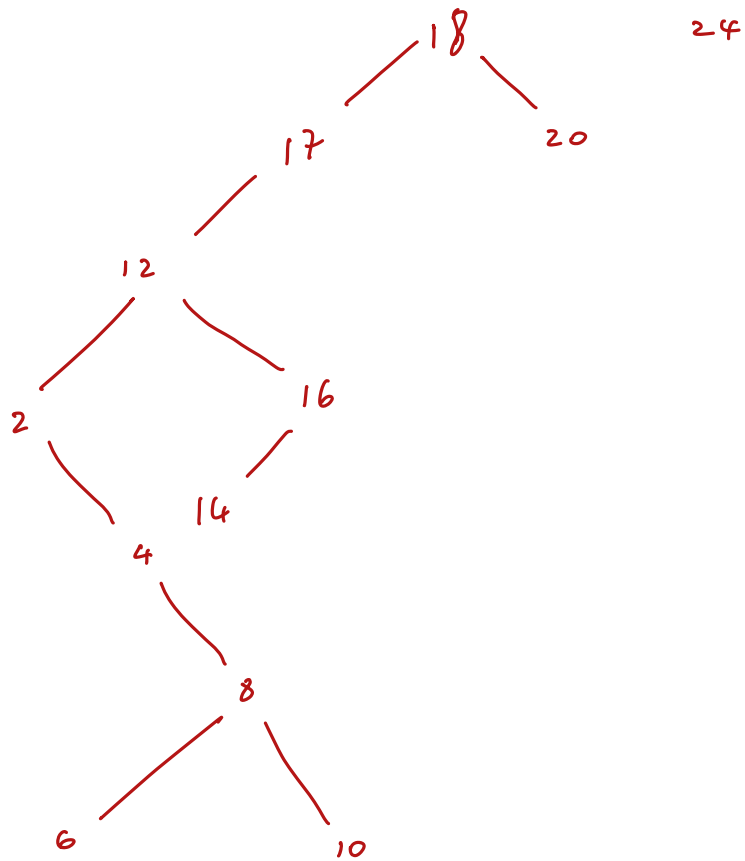
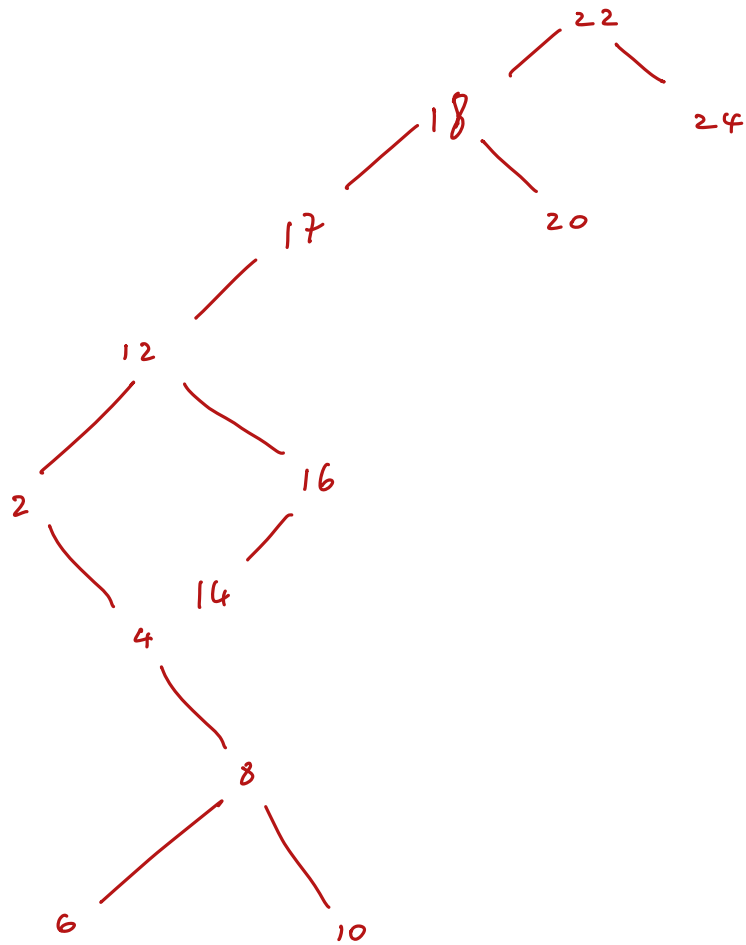


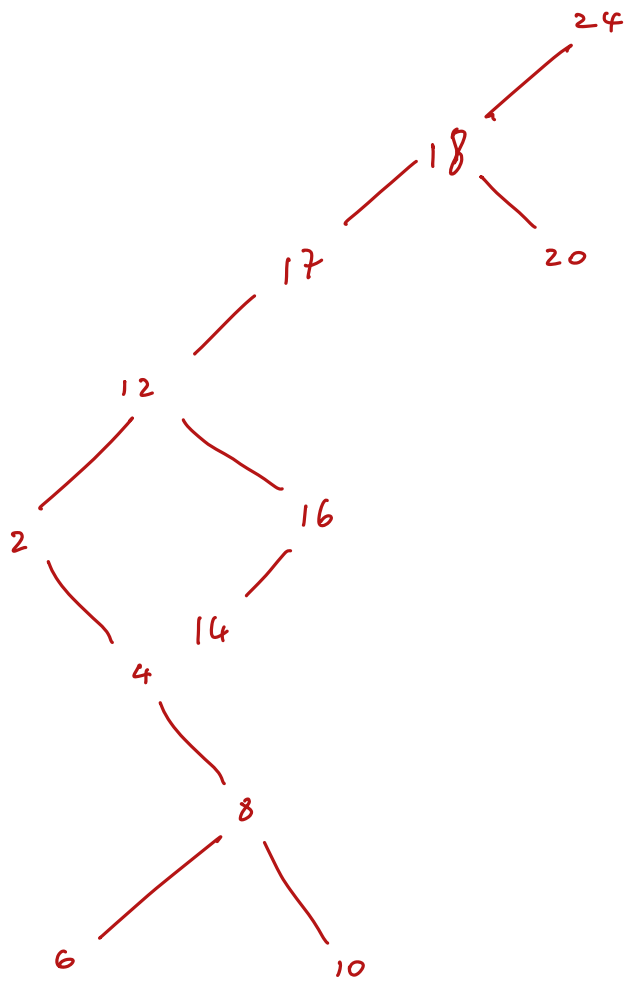
Ø



24

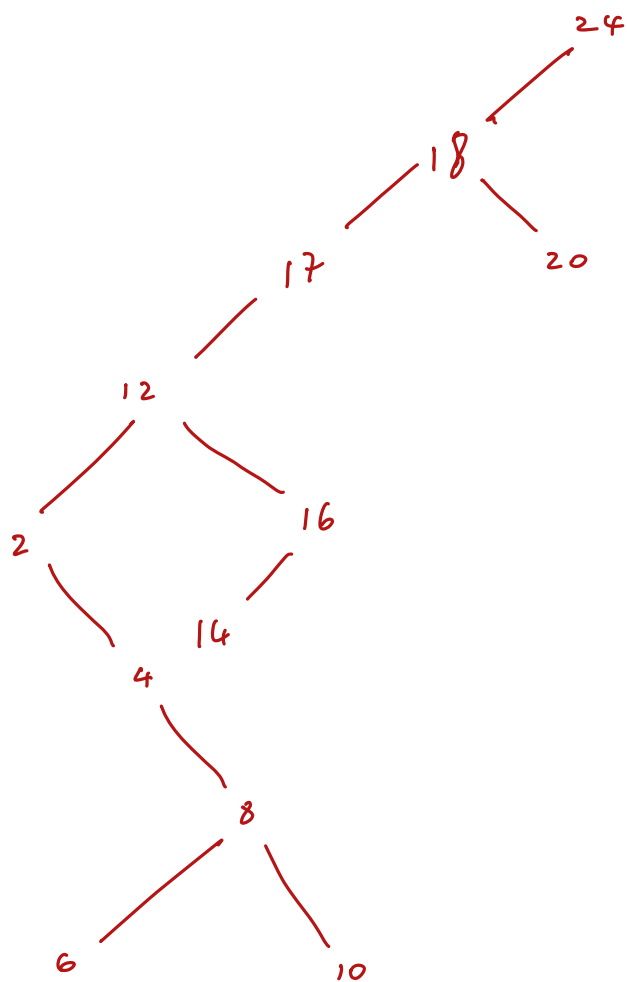
RIASSEMBLAGGIO





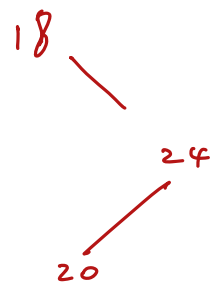
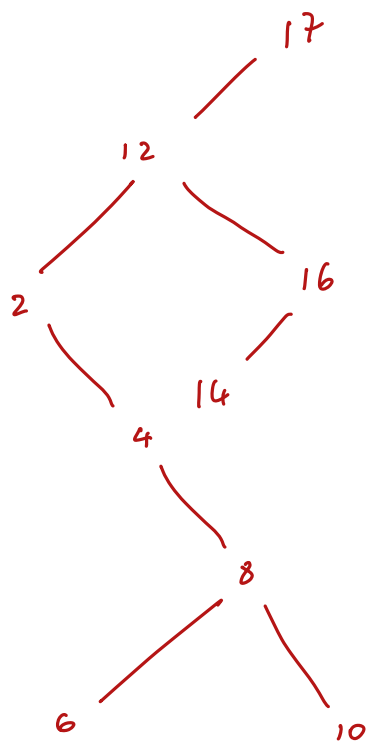
SEARCH(2)

∅

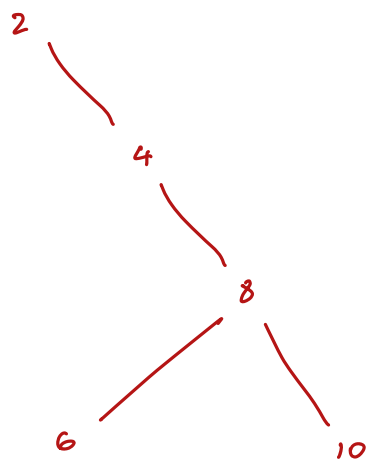


∅

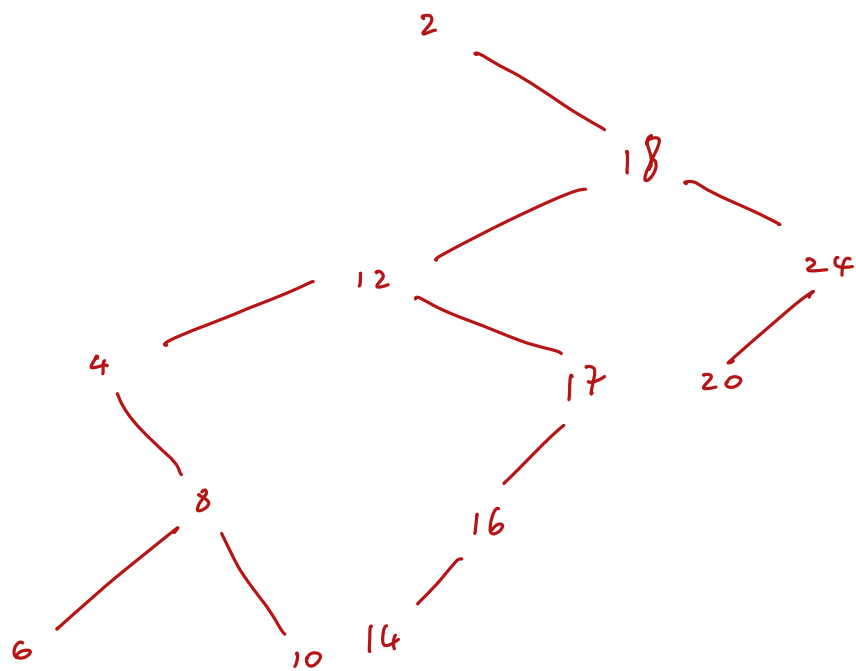
\emptyset



\emptyset



RIASSEMBLAGGIO



ESERCIZIO 1

Utilizzando i tre metodi dell'analisi ammortizzata, si determini il costo ammortizzato per operazione di una sequenza di n operazioni, ove il costo c_i dell' i -esima operazione sia dato da

$$c_i = \begin{cases} 3 \cdot i & \text{se } i \text{ è potenza esatta di } 4 \\ 5 & \text{altrimenti.} \end{cases}$$

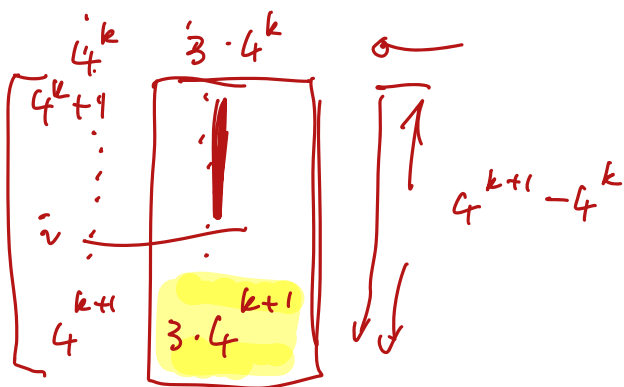
| | |
|----------|---------------|
| 1 | 3 |
| 2 | 5 |
| 3 | 5 |
| 4 | $3 \cdot 4$ |
| 5 | 5 |
| 6 | 5 |
| 7 | 5 |
| 8 | 5 |
| \vdots | \vdots |
| 15 | 5 |
| 16 | $3 \cdot 4^2$ |

$$\frac{3 \cdot 4^{k+1}}{4^{k+1} - 4^k} = \frac{3 \cdot 4}{4 - 1} = \textcircled{4}$$

$$\Phi_i = \begin{cases} 0 & \text{se } i = 0 \\ \textcircled{4} (i - 4^{\lfloor \log_4 i \rfloor}) & \text{se } i > 0 \end{cases}$$

\Uparrow

$$\boxed{\Phi_i \geq 0}$$



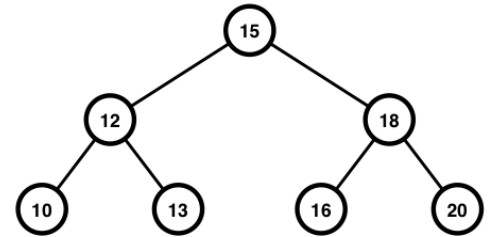
— 4
— 4
:
:
— 4

ESERCIZIO 2

- (a) Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up, quindi si eseguano nell'ordine dato le seguenti operazioni su uno splay tree inizialmente vuoto:
- INSERT 10, 7, 9, 12, 8, 11
 - SEARCH 9
 - DELETE 11
- (b) Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig*, nonché l'operazione di assemblaggio finale, in un splay tree di tipo top-down.

ESERCIZIO 3

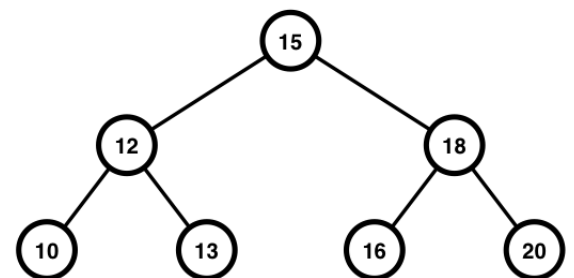
- (a) Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up. Quindi si eseguano nell'ordine dato le seguenti operazioni sullo splay tree a lato:
- SEARCH 13, 20
 - INSERT 17
 - DELETE 16



- (b) Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig*, nonché l'operazione di assemblaggio finale, in un splay tree di tipo top-down.

ESERCIZIO 4 (Splay trees)

- (a) Si descrivano le operazioni di SPLAY, INSERT e DELETE in uno splay tree di tipo bottom-up. Quindi si eseguano, nell'ordine dato, le seguenti operazioni sullo splay tree a lato:
- INSERT 9, 14, 17, 21
 - DELETE 16
 - SEARCH 13, 20



ESERCIZIO 2 (Splay trees)

Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up.

Quindi si eseguano nell'ordine dato le seguenti operazioni su uno splay tree la cui configurazione iniziale è quella di un albero binario completo contenente le 12 chiavi $\{3i : 1 \leq i \leq 12\}$:

- SEARCH 21, 3, 36
- INSERT 25
- DELETE 30
- SEARCH 3

Nota bene: Si ricorda che un albero binario si dice *completo* quando tutti i suoi livelli, con al più l'eccezione dell'ultimo, sono completi e tutti i nodi nell'ultimo livello si trovano il più a sinistra possibile.

ESERCIZIO 2 (Splay trees)

Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up.

Quindi si eseguano nell'ordine dato le seguenti operazioni su uno splay tree la cui configurazione iniziale è quella di un albero binario completo contenente le 10 chiavi $\{4i : 1 \leq i \leq 10\}$:

- SEARCH 20, 40
- DELETE 24
- INSERT 30

Nota bene: Si ricorda che un albero binario si dice *completo* quando tutti i suoi livelli, con al più l'eccezione dell'ultimo, sono completi e tutti i nodi nell'ultimo livello si trovano il più a sinistra possibile.

ESERCIZIO 5 (Splay trees)

Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up.

Quindi si eseguano nell'ordine dato le seguenti operazioni su uno splay tree la cui configurazione iniziale è quella di un albero binario completo contenente le 10 chiavi 12, 7, 9, 15, 14, 2, 1, 10, 6, 5:

- SEARCH 7, 15
- DELETE 10
- INSERT 13

Nota bene: Si ricorda che un albero binario si dice *completo* quando tutti i suoi livelli, con al più l'eccezione dell'ultimo, sono completi e tutti i nodi nell'ultimo livello si trovano il più a sinistra possibile.

ESERCIZIO 5

Si descrivano le operazioni di *zig-zag*, *zig-zig* e *zig* in uno splay tree di tipo bottom-up.

Quindi si eseguano nell'ordine dato le seguenti operazioni su uno splay tree la cui configurazione iniziale è quella di un albero binario completo contenente le 9 chiavi $\{2i : 1 \leq i \leq 9\}$:

- SEARCH 6, 14, 10
- INSERT 13
- DELETE 8
- SEARCH 18

Nota bene: Si ricorda che un albero binario si dice *completo* quando tutti i suoi livelli, con al più l'eccezione dell'ultimo, sono completi e tutti i nodi nell'ultimo livello si trovano il più a sinistra possibile.