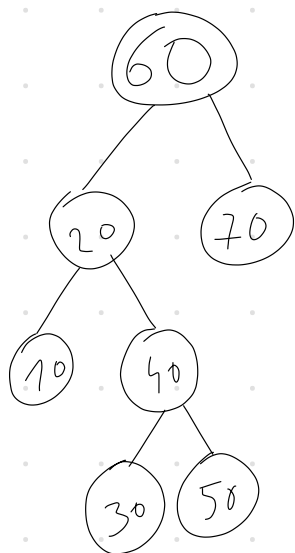


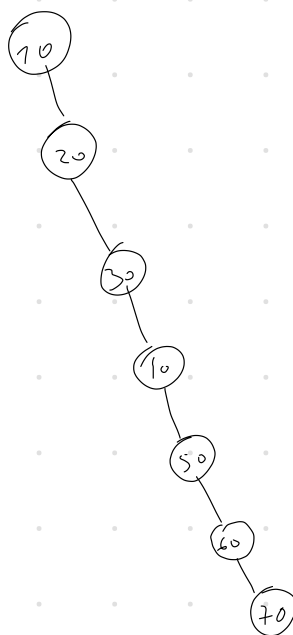
1a



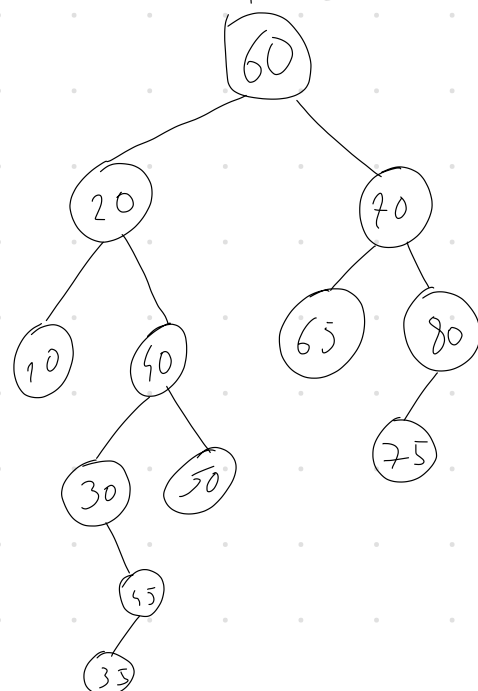
1b

 $h = 4$ 

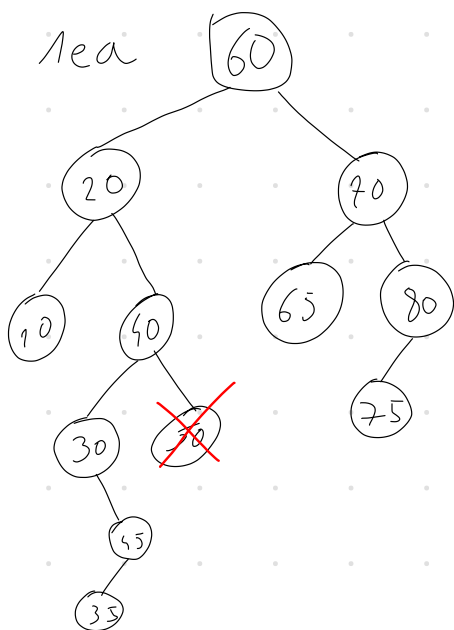
1c



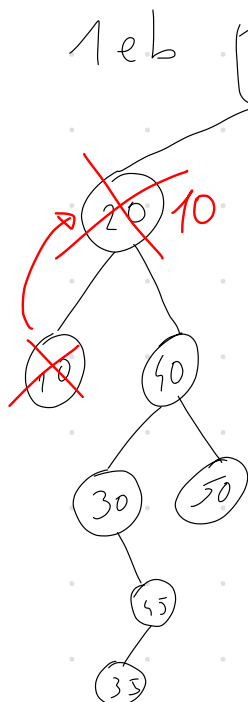
1d



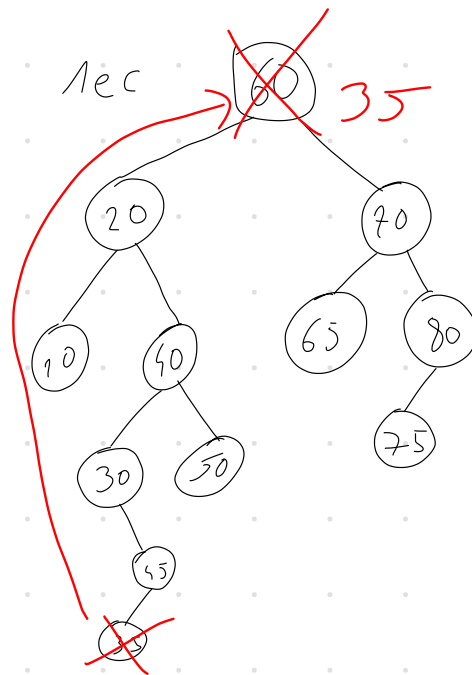
1ea



1eb

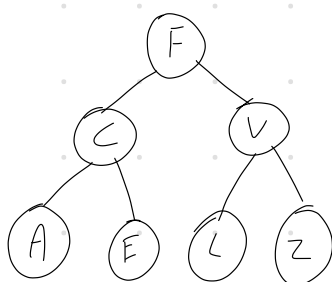


1ec

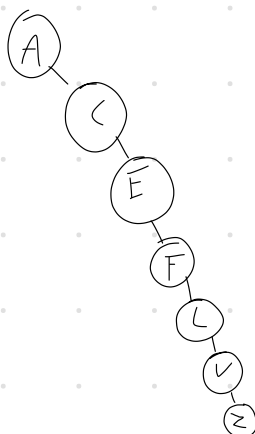


2 Ja, bijvoorbeeld als men na vraag 1ec, 60 terug zou toevoegen.

3. Min



3 Max



4a - Preorder: 60, 40, 30, 20, 10, 50, 70

- Inorder: 10, 20, 30, 40, 50, 60, 70

- Postorder: 10, 20, 30, 50, 40, 70, 60

4b - Preorder: 1, 7, 2, 6, 3, 5, 4

- Inorder: 1, 2, 3, 4, 5, 6, 7

- Postorder: 4, 5, 3, 6, 2, 7, 1

- 5.
- a. Een gesorteerde array doorzoeken gebeurt best aan de hand van een recursief algoritme dat steeds ongeveer de helft van de lijst blijft uitsluiten tot het juiste element het enige overgebleven element is. Alle cases duren even lang.
  - b. Deze enkelvoudig gelinkte gesorteerde ketting doorzoeken kan alleen maar door element per element de ketting af te zoeken en steeds te checken of we er al zijn.
    - Best case: het gezochte element stond eerst, er is 1 zoekactie nodig
    - Worst case: het gezochte element stond laatst, er zijn evenveel zoekacties nodig als er elementen in de lijst staan.
  - c. De efficiëntie van het doorzoeken van een binaire zoekboom hangt heel hard af van hoe de zoekboom opgesteld is.
    - Best case: het gezochte element is de wortel, er is 1 zoekactie nodig
    - Worst case: er zijn evenveel zoekacties nodig als er elementen in de boom staan, bijvoorbeeld wanneer het gezochte element het meest rechts staat