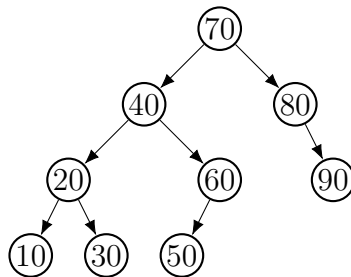


Het tentamen bevat vier opgaven. Graag elke opgave op een nieuwe pagina beginnen. Pseudo-code mag do/od gebruiken of er meer als C++ uitzien, dat is niet belangrijk. Geef steeds voldoende uitleg. Succes.

1. Diverse onderwerpen.
  - a) (i) Beschrijf de abstracte datastructuur *Priority Queue*: wat wordt opgeslagen, wat zijn de standaard operaties (en wat doen ze)? (ii) Leg uit hoe een binaire zoekboom gebruikt kan worden om de *Priority Queue* te implementeren.
  - b) (i) Bij een *splay tree* worden sleutels nadat ze gevonden zijn ‘omhoog geroteerd’. Beschrijf de benodigde operaties: *zig-zag* en *zig-zig*.  
(ii) Teken een lineaire boom van zeven knopen, met alleen rechter kinderen. Pas de *splay* operaties toe op de laagste sleutel in de boom.
  - c) (i) Geef een iteratief algoritme voor een in-order (symmetrische) boom-wandeling op een binaire boom met behulp van een stapel. (ii) Als een bepaalde knoop *node* bezocht wordt, welke knopen staan dan op de stapel?
2. a) Beschouw de volgende AVL boom *A* (hieronder). We gaan één sleutel toevoegen. Welke rotaties worden uitgevoerd (waar, richting, enkel/dubbel) als die sleutel gelijk is aan respectievelijk 5, 45, 75, en 85? (Dus steeds ten opzichte van de oorspronkelijke boom *A*. Je hoeft niet de resulterende bomen te tekenen.)



- b) Verwijder de waarde 70 uit boom *A*. Dit kan op twee manieren, laat deze beide zien, met resultaat.
- c) Voor de hoogte  $h$  van een boom tellen we hier het aantal nivo's van knopen; de boom hierboven heeft bijvoorbeeld hoogte 4.
  - (i) Bereken: een AVL boom van hoogte  $2k$  of  $2k-1$  heeft de eerste  $k$  nivo's gevuld.
  - (ii) Leg uit de operaties toevoegen en verwijderen met een AVL boom logaritmische complexiteit hebben.

3. Ga in deze opgave uit van min-heaps.
- a) Geef de definitie van een *binary heap*, en beschrijf de twee basis-operaties *bubble-up* en *trickle down*.
  - b) (i) Illustreer hoe de priority queue operatie *DeleteMin* wordt uitgevoerd, aan de hand van de volgende heap: [8, 14, 20, 28, 18, 26, 30, 48, 34, 60, 24, 40].  
(ii) Hetzelfde voor *Insert(10)* met dezelfde heap.
  - c) Wat is de complexiteit van het samenvoegen van twee priority queues als deze zijn gerepresenteerd met een *binary heap*? Idem voor de *leftist tree* representatie. (Geef kort uitleg hoe dat samenvoegen gebeurt in de respectievelijke representaties.)
4. We bekijken Ziv-Lempel-Welsh codering.
- a) [ *is vervallen, was verkeerd geformuleerd* ]  
Hieronder heeft het alfabet drie letters a, b, c. Letter a krijgt code 1.  
Geef telkens welke letters ‘geleerd’ worden, en de uiteindelijke codeboom.
  - b) Codeer abbc abcb bbca – spaties staan hier alleen voor de leesbaarheid
  - c) Decodeer 2 4 1 1 4 8 3.