

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.

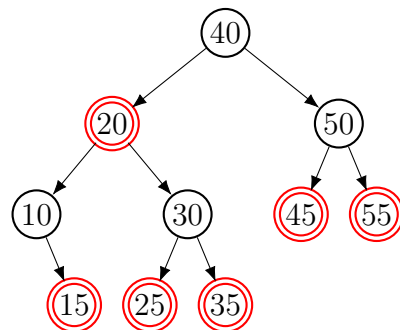
1. a) (i) Geef een iteratief algoritme voor een inorder (symmetrische) boom-wandeling op een (standaard) binaire boom met behulp van een stapel.
(ii) Als een bepaalde knoop **node** bezocht wordt, welke knopen staan dan op de stapel?

We gaan hieronder uit van een gewone bedrade boom, met symmetrische rechterdraden.

- b) Hoe vinden we de opvolger van een gegeven knoop **node** in de boom?
- c) Als we in een bedrade boom een nieuw blad toevoegen moeten ook de draden aangepast worden. Bespreek welke draad (of draden) veranderen, en hoe. Met een plaatje graag.

2. a) Wat zijn de definiërende eigenschappen van de rood-zwart boom (*red-black tree*)?

Gegeven is de volgende rood-zwart boom, waarbij ‘rode’ knopen een extra cirkel hebben gekregen.



- b) Welke boom ontstaat als we hieraan achtereenvolgens toevoegen 12, 32 en 52. Benoem de gebruikte operaties en geef relevante tussenresultaten.
- c) (i) Geef een voorbeeld van een rood-zwart boom waar bij toevoegen van een sleutel twee opeenvolgende flag-flips plaatsvinden.
(ii) Leg uit dat er bij toevoegen nooit twee opeenvolgende (enkele/dubbele) rotaties nodig zijn.

3. Ga in deze opgave uit van max-heaps.
- a) Beschrijf de abstracte datastructuur *priority queue*: wat wordt opgeslagen, wat zijn de standaard operaties (en wat doen ze)?
 - b) Geef de definitie van een *binary heap*. Beschrijf hoe de twee basis-operaties *bubble up* en *trickle down* gebruikt worden om een *priority queue* te implementeren.
 - c) De heap-structuur is gemotiveerd door sorteren: *heap-sort*. Bekijk het gedeeltelijk gesorteerde array $[45, 40, 25, 30, 35, 10, 20, 15, 50, 55]$ met tien elementen. De laatste twee elementen zijn reeds gesorteerd.
 - (i) Welke eigenschap hebben de eerste acht elementen? Controleer dit.
 - (ii) Breng stapsgewijs twee extra elementen over naar het gesorteerde gedeelte, en hou de overige elementen weer op orde. Leg uit wat u doet.
4. De methode van Knuth-Morris-Pratt wordt gebruikt om een patroon P in een tekst T te zoeken. Daartoe worden *failure-links* opgesteld.
- a) Geef een efficiënt algoritme dat de *failure-links* voor een patroon bepaalt.
 - b) Bepaal de *failure-links* voor het patroon $P = \text{AABA ABAB A}$. (Spaties voor de leesbaarheid).
 - c) We zoeken naar P in de tekst $T = \text{BBAA BBAB BABB BBAB BBAB}$. Geef nauwkeurig aan hoe het zoeken volgens de KMP-methode gebeurt. Welke letters worden telkens met elkaar vergeleken?