

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. Veel succes.

1. We kijken naar een symmetrisch bedrade boom (*inorder threads*) met alleen draden naar rechts. (De draden zijn permanent, en worden van takken onderscheiden door een boolean `RightThread` in elke knoop.)

We onderzoeken in deze opgave welke boomwandelingen we kunnen maken met symmetrische bedrading. Gebruik in de antwoorden pseudo-code en een plaatje.

- a) – Hoe vinden we de eerste knoop in in-orde volgorde uitgaande van de wortel `R`?
– Hoe vinden we de in-orde opvolger van een knoop `curr`?

De volgende onderdelen kijken naar andere boomwandelingen, maar nog steeds in een symmetrisch bedrade boom!

- b) – Hoe vinden we de eerste knoop in pre-orde volgorde uitgaande van de wortel `R`?
– Hoe vinden we de pre-orde opvolger van een knoop `curr`?
- c) – Hoe vinden we de eerste knoop in post-orde volgorde uitgaande van de wortel `R`?
– Laat met een eenvoudig voorbeeld zien dat we *niet* van elke knoop `curr` de post-orde opvolger kunnen bereiken vanuit `curr` als we alleen takken en draden mogen volgen.

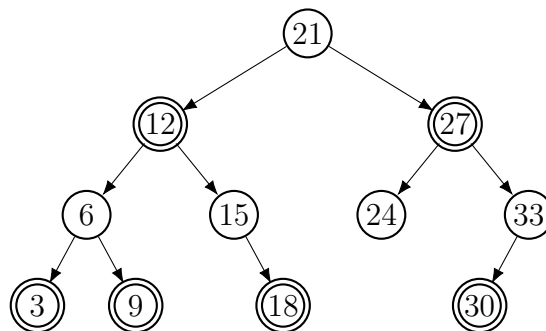
2. a) Wat is de definitie van de rood-zwart boom (*red-black tree*) ?

- b) Er zijn twee operaties die de rood-zwart boom herstructureren: *flag-flip* en rotatie.

(i) Kan het toevoegen van een enkel element meerdere flag-flips veroorzaken?

(ii) Kan het toevoegen van een enkel element meerdere rotaties veroorzaken?

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



- c) Welke boom ontstaat als we achtereenvolgens aan R de volgende getallen toevoegen: 2, 16, 28.

Geef duidelijk de tussenstappen weer.

(Begin steeds opnieuw met de oorspronkelijke boom R .)

- d) Op welke voor de hand liggende manier kunnen we waarde 12 uit de boom verwijderen?

-
3. De ADT *Priority Queue* heeft als operaties Insert, FindMin en DeleteMin.
- a) (i) Geef de definitie van *binary heaps*, en beschrijf hun twee basisoperaties *Trickle Down* en *Bubble Up*.
(ii) Hoe implementeren we daarmee een Priority Queue?
 - b) (i) Geef de definitie van *leftist trees*, en beschrijf hun basisoperatie ‘ritsen’ (*Zip*).
(ii) Hoe implementeren we daarmee een Priority Queue?
 - c) Priority Queues kunnen worden samengevoegd, de operatie heet soms *Meld*.
(i) Hoe implementeren we Meld voor de binary heap? Wat is de complexiteit?
(ii) Hoe implementeren we Meld voor de leftist tree? Wat is de complexiteit?
4. a) Zowel Huffman als Ziv-Lempel-Welsh zijn compressie algoritmen. Beschrijf de boomstructuur van de bijbehorende codes.
- b) Wat is de complexiteit van het algoritme van Huffman voor n symbolen en hun frequenties? Leg uit.
- c) Gegeven is een string met symbolen A, C, G, N, T en X, met frequenties 18, 15, 14, 7, 20 en 8.
(i) Pas het algoritme van Huffman toe, en geef het resultaat.
(ii) Hoeveel bits zijn nodig om de tekst te coderen?
- d) Het algoritme van Huffman maakt soms keuzes die een andere code opleveren, maar met dezelfde efficiëntie. Is er een mogelijke codering bij de frequenties hierboven, die niet door Huffman gevonden kan worden, maar even goed is?