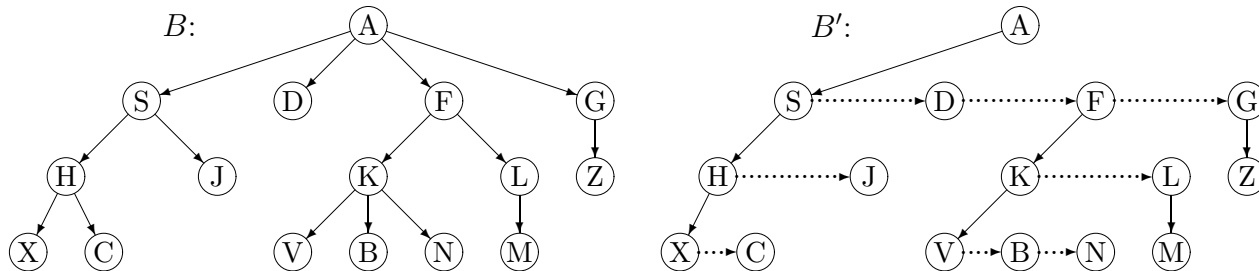


Gevraagde functies en programma's mogen in pseudo-code gegeven worden.

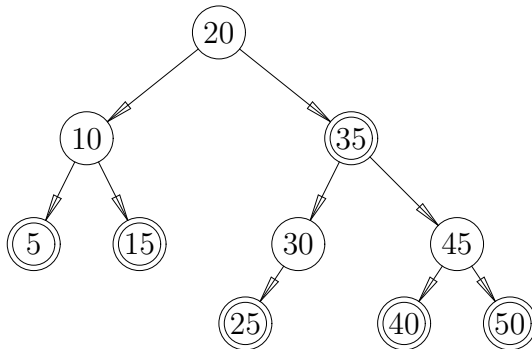
Geef steeds voldoende uitleg.

- 1) Bomen met willekeurige graad van de knopen kunnen gerepresenteerd worden met de EersteKind-RechterBroer representatie ('EK-RB'). Hieronder als illustratie een boom B in de klassieke representatie, en de overeenkomstige boom B' in EK-RB representatie.



- Geef een algoritme dat een pre-orde wandeling uitvoert in een boom in EersteKind-RechterBroer representatie. Dit algoritme dient gebruik te maken van een stapel, zonder recursie.
Leg (in een toelichting) uit welke knopen op de stapel gezet worden.
- Om een boomwandeling in de EersteKind-RechterBroer representatie te kunnen uitvoeren zonder stapel of recursie, willen we de representatie uitbreiden met draden. Doe een voorstel voor het bedraden van deze bomen. Leg uit hoe deze draden in het algemeen lopen, en teken deze draden in de boom B' hierboven (neem het plaatje over in uw uitwerkingen).
- Geef een algoritme dat een systematische wandeling uitvoert in de door u bedrade bomen in EersteKind-RechterBroer representatie. Neem aan dat elke knoop voorzien is van een tag-veld dat draden markeert.
Zorg er voor dat het algoritme zowel voor pre-orde als post-orde bruikbaar is, en geef de plek van het pre-orde en post-orde bezoek in de code aan.

- 2) a. Wat zijn de kenmerkende eigenschappen van de rood-zwart boom (*red-black tree*)? Deze boom is op het college gepresenteerd als implementatie van de B-boom van orde $m = 4$.
- b. Gegeven is de volgende rood-zwart boom, waarbij ‘rode’ knopen een extra cirkel hebben gekregen.



Welke boom ontstaat als we hieraan toevoegen

- eerst 55 en vervolgens 7.
- eerst 7 en vervolgens 55.

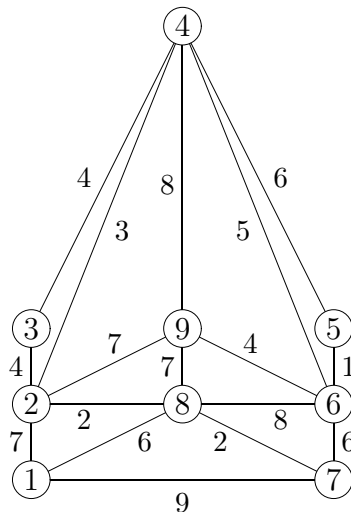
Benoem de achtereenvolgende operaties en geef relevante tussenresultaten.

- c. Hoeveel knopen (rode en zwarte samen) heeft een rood-zwart boom van hoogte h maximaal? En minimaal?

De hoogte van de boom wordt hier gemeten in het aantal knopen op het langste pad van wortel naar blad.

Voor het minimale aantal hoeft u alleen even [óf oneven, naar keuze] hoogtes te bespreken.

- 3) Met het algoritme van Prim kunnen we voor samenhangende, ongerichte grafen met gewichten op de takken een minimale opspannende boom bepalen.
- Geef een pseudo-code implementatie van het algoritme van Prim.
 - Pas het algoritme van Prim toe op de volgende graaf. Gebruik knoop 1 als beginknoop van het algoritme.



Geef voor elke stap in het algoritme duidelijk aan hoe u tot een bepaalde keuze komt. Als u het plaatje niet een aantal maal wil kopiëren, kunt u de werking van het algoritme ook in tabelvorm weergeven.

- Het algoritme van Dijkstra (voor het bepalen van een boom van kortste afstanden) lijkt sterk op het algoritme van Prim. Op welke plaats(en) in de pseudo-code van onderdeel (a) wijkt het algoritme van Dijkstra af? Geef ook de afwijkende code.
- Gegeven een ongerichte graaf met n knopen en e takken, wat is bij een slimme keuze van de gebruikte datastructuren de complexiteit van het algoritme van Prim?
Er worden twee antwoorden verwacht(!) Wat is in elk van die gevallen de 'duurste' operatie in het algoritme? Licht uw antwoord toe.

- 4) a. Bij het algoritme van Knuth-Morris-Pratt berekenen we *failure links* voor ieder van de posities in het patroon dat we zoeken.
Wat weten we van het patroon als gegeven is dat de failure link van positie k gelijk is aan r ?
- b. Bereken de failure links voor het patroon $P = \text{abacaaba}$.
- c. We zoeken naar P in de tekst $T = \text{abab caab aaab acab acaa baba}$ (hier staan de spaties louter voor de leesbaarheid). Geef nauwkeurig aan hoe het zoeken volgens het KMP-algoritme gebeurt. Welke letters worden telkens met elkaar vergeleken?
- d. Bepaal de failure links voor het patroon $P = \text{tentamen}$.
- e. Het naïeve algoritme om patronen in een tekst te zoeken werkt als volgt: we leggen het patroon aan het begin langs de tekst en we vergelijken patroon en tekst letter voor letter. Wanneer er verschil optreedt, schuiven we het patroon één letter verder en we beginnen opnieuw.
- Stel nu dat we in een willekeurige Nederlandse tekst zowel met het KMP-algoritme, als met het naïeve algoritme op zoek gaan naar het patroon **tentamen**. Welk algoritme zal dan het snelst werken? Verwacht u een groot verschil in efficiëntie tussen de beide algoritmes? Verklaar uw antwoord.