1a. Een queue is een lineaire lijst waar de elementen aan de ene zijde worden toegevoegd en aan de andere zijde bekeken en verwijderd. Op die manier krijgen we FIFO gedrag: first-in-first-out.

Standaard operaties: Init (levert een lege lijst op), IsEmpty (wanneer de lijst leeg is), EnQueue/Put (plaatst een element achteraan de rij), DeQueue/Get (verwijdert een element vooraan de rij, ongedefinieerd op een lege lijst), Peek (bekijkt het element vooraan), Size (levert aantal elementen).

Implementaties: een array (op een cyclische manier gebruikt, met index naar eerste en laaste element), een lineaire lijst (met pointers naar het eerste en laaste element; welke kant lopen de pointers van de lijst?)

1b. Verzameling elementen met een prioriteit. Prioriteiten zijn totaal geordend (elk tweetal kan vergeleken worden, meestal gehele of reële getallen). Init, IsEmpty, Insert (voeg element met prioriteit toe), DeleteMax (verwijdert het element met hoogste prioriteit, ongedefinieerd op een lege queue), GetMax (bekijkt dat element).

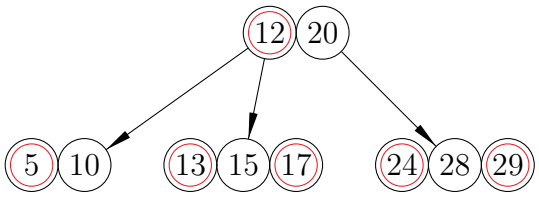
1c. Twee elementen per stap herbalanceren. Zie dictaat.

Een binaire zoekboom (met sleutels geordend op prioriteit) kan op de standaard manier sleutels plaatsen. Het maximale element bevindt zich helemaal rechts.

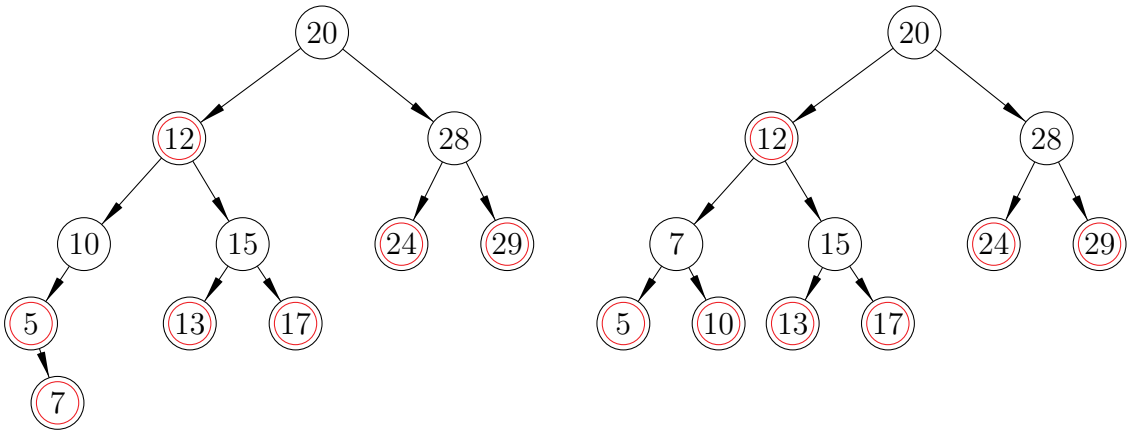
1d. todo. [dictaat?]

2a. Een rood-zwart boom is een binaire zoekboom waarvan de knopen zwart of rood gekleurd zijn. Er gelden de volgende restricties: (a) de wortel is zwart, (b) een rode knoop kan geen rode ouder hebben, (c) de 'zwart hoogte', het aantal zwarte knopen van wortel naar blad, is gelijk voor ieder blad.

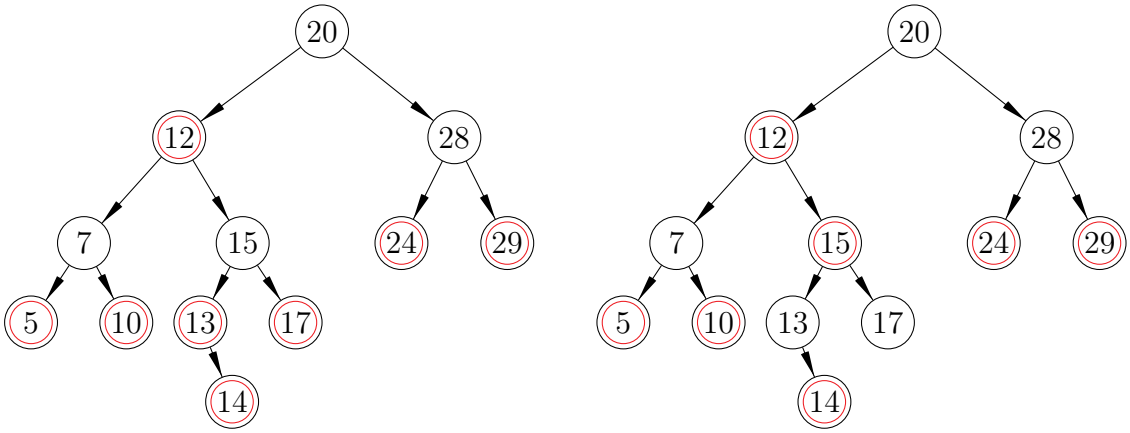
b. Als we de rode knopen naast hun respectievelijke ouders tekenen lijkt de boom al op de bijbehorende B-boom. Technisch hebben we dan een HV-boom (hoewel hier alle horizontale takken verdwenen zijn).

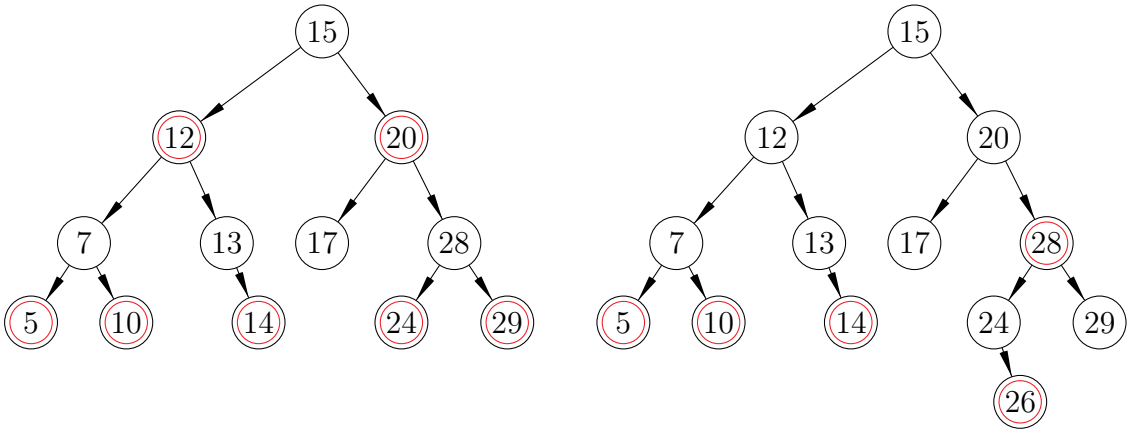


d. Voeg 7 toe, als rood rechter kind van 5. [Dia 1] Rood op rood, oom zwart (null bladeren zijn zwart), links-rechts, herstel door dubbele rotatie naar rechts (bij 10). [Dia 2]



Voeg 14 toe, als rood rechter kind van 13. [Dia 3] Rood op rood, oom rood, flag-flip. Nu is er rood op rood bij 15. [Dia 4] Oom zwart, dubbele rotatie om 20 naar rechts. [Dia 5]





Voeg 26 toe, als rood rechter kind van 24. Rood 24 op rood 28, oom 29 rood, flag-flip. Rood 28 op rood 20, oom 12 rood, flag-flip. Wortel 15 rood, zwart kleuren. [Dia 6] Hiermee is de zwart hoogte van de boom één toegenomen.

3a. Bij perfect hashen wordt een vooraf gekozen verzameling K van sleutels opgeslagen in een tabel, gebruik makend van een hashfunctie zonder botsingen. (Ofwel, er zijn geen synoniemen onder de elementen van K.)

Dit is van toepassing als we een van te voren bepaalde hoeveelheid sleutels willen herken-

nen, zoals keywords in een programmeertaal, of een reeks veelvoorkomende woorden in een taalkundige bron.

b. Stapgrootte 1 leidt ertoe dat bij het voorgaande adres gekeken wordt of deze vrij is. Als de hash-methode een aanleiding geeft om te veronderstellen dat juist die plek een grote kans heeft om bezet te zijn kunnen we beter een grotere stapgrootte kiezen. Voorbeeld sleutels als `hulp1`, `hulp2`, ... Als alternatief kunnen ook die sleutels met een andere adresfunctie uit elkaar gehaald worden.

c. Primaire clustering. Bij lineair hashen neigen er blokken te ontstaan van aaneengesloten bezette adressen. (Dit leidt tot slechte zoekresultaten, omdat soms zo'n blok geheel bezocht wordt voordat we concluderen dat een sleutel niet in de tabel zit.) Grotere blokken groeien ook sneller, omdat alle sleutels die op een adres uit het blok toegevoegd worden uitkomen op het adres net voor het blok.

Secundaire clustering. Bij hashmethoden waarbij synoniemen hetzelfde zoekpad volgen. (Dit is geen zichtbaar 'blok', maar synoniemen zitten elkaar wel in de weg.) Oplossing is het kiezen van een dubbele hashmethode.

d. De adressen (synoniemen zijn duidelijk te herkennen) en de stapgroottes:

K	43	38	7	23	30	25	29	12	40
$K \bmod 11$	10	5	7	1	8	3	7	1	7
$(K \bmod 4) + 1$	4	3	4	4	3	2	2	1	1

We lopen naar links, dat staat expliciet in de opgave. Modulo zorgt ervoor dat de twee uiteinden van de tabel als het ware tegen elkaar liggen.

0	1	2	3	4	5	6	7	8	9	10	
	23		25		38		7	30		43	direct geplaatst, geen synoniemen
	<u>23</u>		<u>25</u>		<u>38</u>	29	<u>7</u>	<u>30</u>		<u>43</u>	h(29)=7, bezet, $p(29) = 2$ dus naar $7-2 = 5, 3, 1, -1 \equiv 10, 8, 6$
12	<u>23</u>		25		38	29	7	30		43	h(12)=1, bezet, p(12)=1, plaatsen op $1-1 = 0$ want vrij
12	23		25	40	<u>38</u>	<u>29</u>	<u>7</u>	30		43	h(40)=7, bezet, p(40)=1, proberen op $7-1 = 6$ en 5, plaats op 4
12	23		25	40	38	29	7	30		43	resultaat

4a. Efficient, dus gebruikmakend van eerdere links van het patrooon. Zie elders.

bd.

k	1	2	3	4	5	6	7	8
$P[k]$	B	B	A	B	B	B	A	B
FLink$[k]$	0	1	2	1	2	3	3	4
FLink$'[k]$	0	0	2	0	0	3	2	0

c. stap 1. Start op positie $T[1]$, met letter $P[1]$. Match voor $P[1 \ldots 3]$. Fout op positie $P[4]$. FLink$[4] = 1$.

stap 2. Start op positie $T[4]$, met letter $P[1]$. Fout op positie $P[1]$. FLink$[1] = 0$.

stap 3. start op eerste positie patroon $P[1]$ op volgende positie tekst $T[5]$. Match voor $P[1 \ldots 5]$. Fout op positie $P[6]$. FLink$[6] = 3$.

stap 4. Start op positie $T[10]$, met letter $P[3]$. Match voor $P[3 \ldots 6]$. Fout op positie $P[7]$. FLink[7] = 3.

stap 5. Start op positie $T[14]$, met letter $P[3]$. Fout op positie $P[3]$. FLink[3] = 2.

stap 6. Start op positie $T[14]$, met letter $P[2]$. Match voor $P[2 \ldots 8]$. Patroon gevonden, op positie $T[14]$.

k	1	2	3	4	5	6	7	8	9				13				17			
$T[k]$	B	B	A	A	B	B	A	B	B	A	B	B	B	B	A	B	B	B	A	B
P	$\underline{B}_1$	$\underline{B}$	$\underline{A}$	B_4																
P				B_1																
P					$\underline{B}_1$	$\underline{B}$	$\underline{A}$	$\underline{B}$	$\underline{B}$	B_6										
P										$\underline{A}_3$	$\underline{B}$	$\underline{B}$	$\underline{B}$	A_7						
P														A_3						
P														$\underline{B}_2$	$\underline{A}$	$\underline{B}$	$\underline{B}$	$\underline{B}$	$\underline{A}$	$\underline{B}_8$

d. Dit kan door het patroon en de failure links van links naar rechts te doorlopen. Zie elders. Nieuwe links zie boven.