

antwoorden proeftentamen datastructuren december 2007

Opgave 1.

- (a) We implementeren de operaties **enqueue** en **dequeue** van het ADT voor queues met twee stacks S_1 en S_2 .

De operatie **enqueue**(o, Q) voegt een element toe aan het eind van de queue Q . Deze operatie wordt geïmplementeerd door **push**(o, S_1). Dit is in $O(1)$.

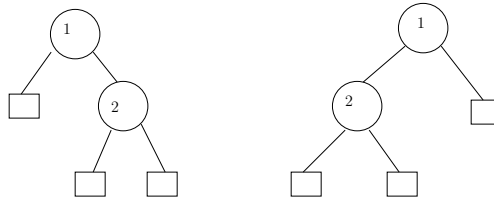
De operatie **dequeue**(Q) geeft en verwijdert het element vooraan in de queue Q . Deze operatie wordt als volgt geïmplementeerd: Als S_2 niet leeg, dan **pop**(S_2).

Als S_2 leeg, dan **push**(**pop**(S_1), S_2) totdat S_1 leeg is, dan **pop**(S_2). Dit is gemiddeld in $O(1)$.

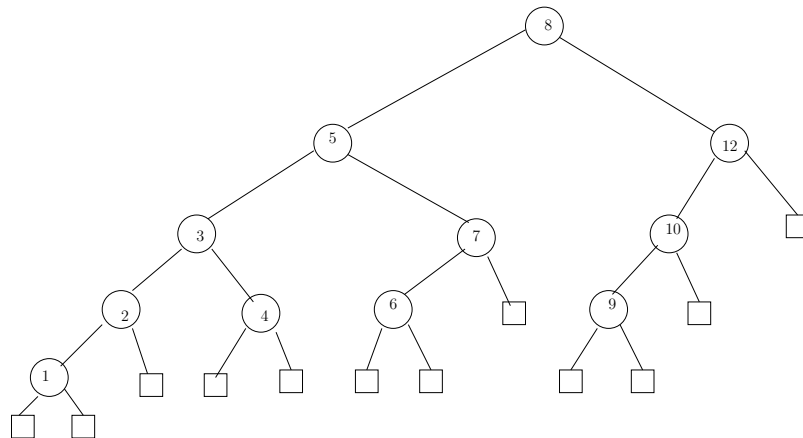
- (b) We voegen eerst 1 en vervolgens 2 toe aan een initeel lege binaire zoekboom. Dit levert de linkerboom in het plaatje.

We voegen eerst 2 en vervolgens 1 toe aan een initeel lege binaire zoekboom. Dit levert de rechterboom in het plaatje.

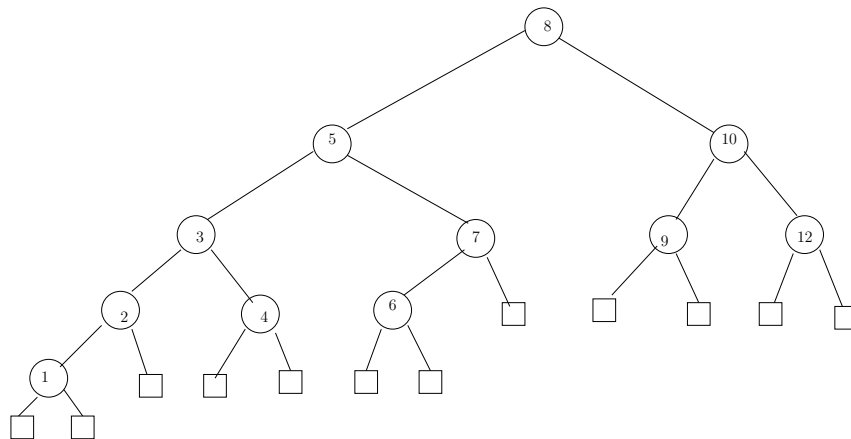
De linker- en rechterboom zijn verschillend.



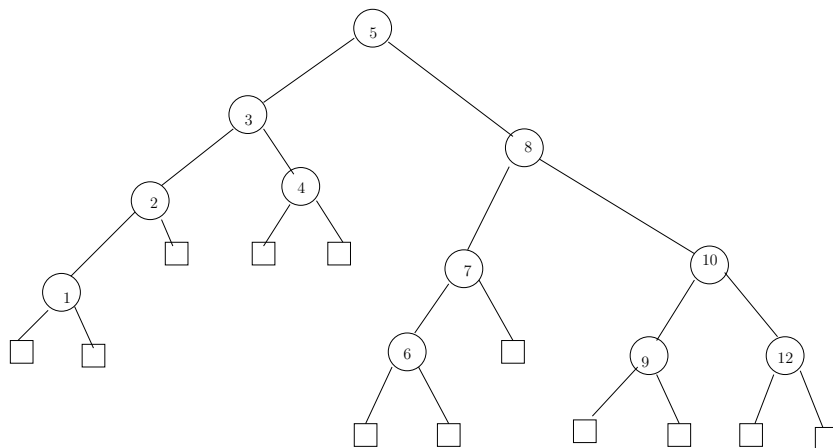
- (c) Om te beginnen verwijderen we de knoop met label 11 volgens de knoopverwijderen procedure voor binaire zoekbomen. Dit levert de volgende binaire zoekboom:



Dit is geen AVL-boom. De onderste knoop die niet in balans is is de knoop met label 12. Een rotatie van de knopen met labels 12, 10 en 9 levert de volgende boom:



Dit is nog steeds geen AVL-boom. De volgende knoop die niet in balans is is de root. Een rotatie van de knopen met labels 8, 5, en 3 levert de volgende boom:



- (d) Het volgende algoritme geeft **true** terug als de subboom van een binaire boom b vanaf knoop v een echte binaire boom is, en **false** anders.

algoritme PBT(b,v):

```

if isExternal( $b,v$ )
  then return true

```

```

else if PBT(b,leftChild(v))=true
    then if PBT(b,rightChild(v))=true
        then return true
    else return false

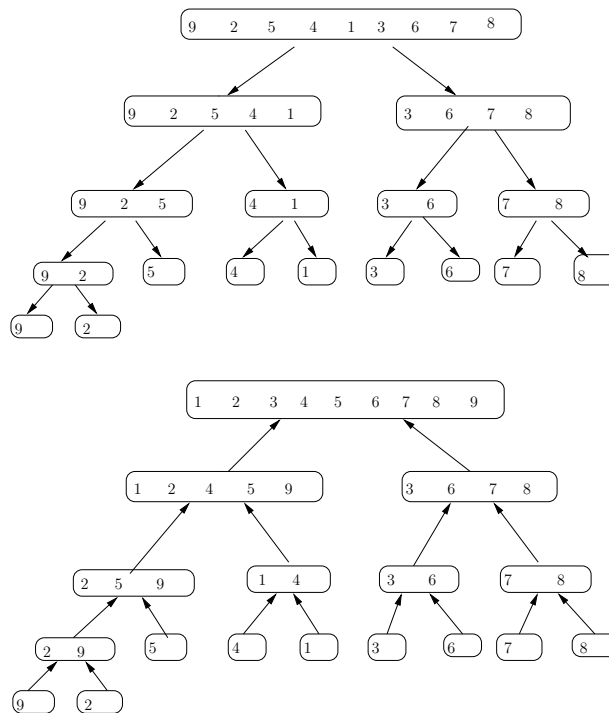
```

We gaan ervan uit dat de operaties `leftChild` en `rightChild` een error opleveren (en dus geen `true`) als het linker- of rechter-kind niet bestaat.

Nu geeft $\text{PBT}(b, \text{root}(b))$ als output `true` precies als b een echte binaire boom is.

Opgave 2.

(a)



(b) Er zijn ten minste 10 en ten hoogste 29 vergelijkingen nodig.

(c) We sorteren de rij:

(1, 3) (2, 2) (1, 1)

Eerst sorteren we de belangrijkste (dus eerste) component. Dit levert:

(1, 3) (1, 1) (2, 2)

En vervolgens de tweede component; dit levert:

$$(1, 1) \quad (2, 2) \quad (1, 3)$$

Dit is niet goed, want $(1, 3)$ hoort voor $(2, 2)$ te komen.

- (d) Eerst representeren we de getallen in het getalsysteem met basis 4. Dit levert de volgende rij:

$$22 \quad 13 \quad 33 \quad 01 \quad 11 \quad 23$$

We sorteren vervolgens de tweede component (stabiel). Dit levert:

$$01 \quad 11 \quad 22 \quad 13 \quad 33 \quad 23$$

En vervolgens sorteren we de eerste component (stabiel). Dit levert:

$$01 \quad 11 \quad 13 \quad 22 \quad 23 \quad 33$$

Opgave 3.

- (a) De gewogen waarden zijn:

item	gewogen waarde
a	3
b	5
c	3

We nemen dus item b en $\frac{1}{3}$ van item a. (Of: item b en $\frac{1}{2}$ van item c.)

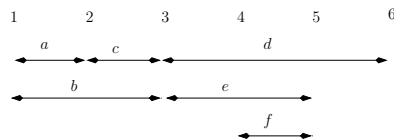
- (b) Het brute-force algoritme bekijkt alle mogelijke deelverzamelingen. Dit algoritme is in $O(2^n)$, met n het aantal elementen in de verzameling.

Het dynamic-programming algoritme is in $O(nW)$, met n het aantal elementen van de verzameling en W het maximum totaal gewicht.

Het dynamic-programming algoritme is beter dan het brute-force algoritme als er geldt $nW < 2^n$. In dit geval: als $W < 2^5$.

Opgave 4.

- (a)



- (b) Een greedy algoritme: Kies steeds uit de taken die nog gescheduled kunnen worden een taak met kleinste eind-tijd.

Opgave 5.

- (a) Met $\emptyset$ als verzameling toegestane tussenpunten, en volgorde $u \quad v \quad w$:

0	2	1
∞	0	∞
1	4	0

Met $\{u\}$ als verzameling toegestane tussenpunten:

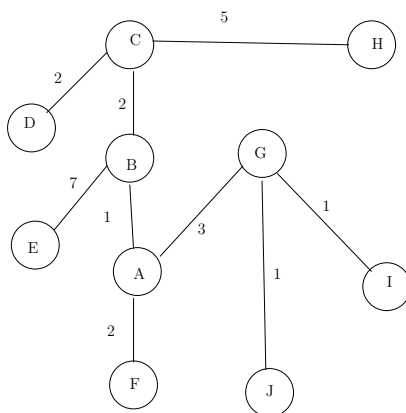
0	2	1
∞	0	∞
1	3	0

De uitbreidingen tot $\{u, v\}$ en $\{u, v, w\}$ als toegestane tussenpunten leveren geen nieuwe informatie.

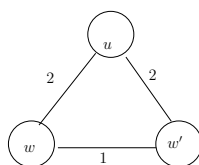
- (b) We voeren het algoritme van Prim-Jarník uit; een gewicht in **boldface** geeft aan dat de bijbehorende kant definitief gekozen is.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
null									
0	1	∞	∞	∞	2	3	∞	∞	5
null	(A, B)				(A, F)	(A, G)			(A, J)
0	1	2	∞	7	2	3	∞	∞	5
null	(A, B)	(B, C)		(B, E)	(A, F)	(A, G)			(A, J)
0	1	2	2	7	2	3	5	∞	5
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)		(A, J)
0	1	2	2	7	2	3	5	∞	5
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)		(A, J)
0	1	2	2	7	2	3	5	∞	4
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)		(F, J)
0	1	2	2	7	2	3	5	1	4
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)	(G, I)	(F, J)
0	1	2	2	7	2	3	5	1	1
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)	(G, I)	(G, J)
0	1	2	2	7	2	3	5	1	1
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)	(G, I)	(G, J)
0	1	2	2	7	2	3	5	1	1
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)	(G, I)	(G, J)
0	1	2	2	7	2	3	5	1	1
null	(A, B)	(B, C)	(C, D)	(B, E)	(A, F)	(A, G)	(C, H)	(G, I)	(G, J)

We vinden dus de volgende minimale opspannende boom:



(c) Bekijk de volgende gewogen ongerichte graaf:



De boom met de kortste paden (links in het volgende plaatje) is geen minimale opspannende boom. Rechts in het plaatje staat een minimale opspannende boom; die heeft totaalgewicht 3.



Opgave 6.

(a) Het brute-force pattern-matching algoritme heeft voor het geval

$$\begin{aligned} P &= aab \\ T &= a^{1000}b \end{aligned}$$

$999 \times 3 = 2997$ vergelijkingen nodig.

(b) In het algemeen heeft het brute-force pattern-matching algoritme in het ongunstigste geval $(n - m + 1)m$ vergelijkingen nodig.

(c) Het brute-force pattern-matching algoritme is in $O(nm)$.

(d) De failure functie voor $P = aaa$ is:

j	0	1	2
$P[j]$	a	a	a
$f(j)$	0	1	2

Een uitvoering van het Knuth-Morris-Pratt algoritme met patroon $P = aaa$ en tekst $T = babaabaaa$:

```

b  a  b  a  a  b  a  a  a
↑
a  a  a
   |  ↑
   a  a  a
       ↑
       a  a  a
           |  |  ↑
           a  a  a
               ↑
               a  a  a
                   ↑
                   a  a  a
                       |  |  |
                       a  a  a

```