

Bomen en grafen

Les 6

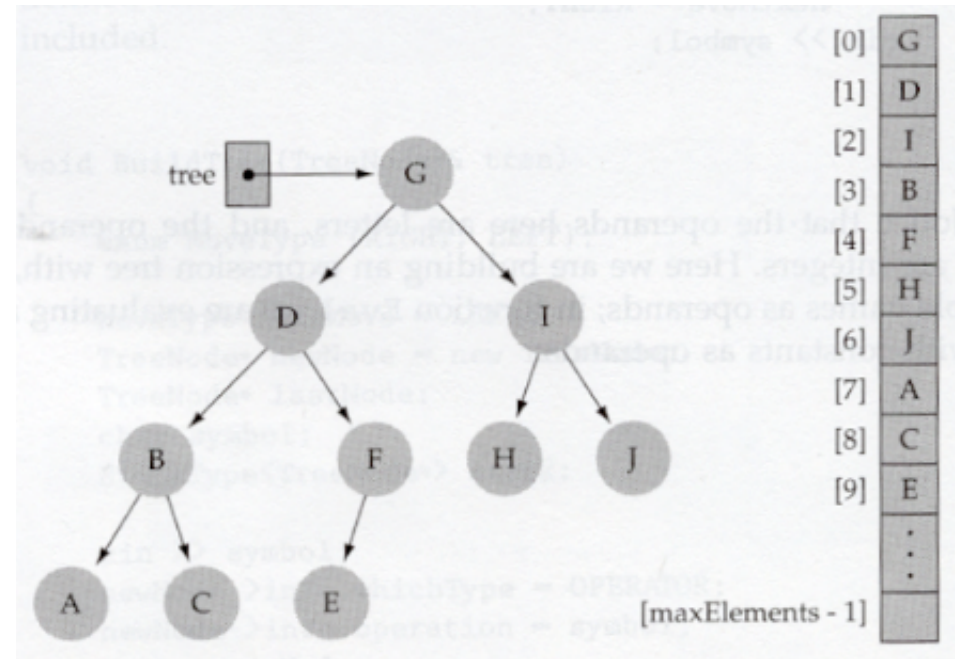
Array implementatie

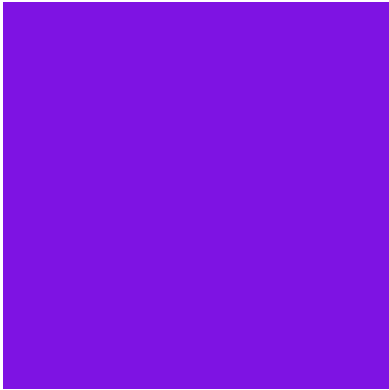
- Het is mogelijk om een boom intern als een lineaire datastructuur voor te stellen.
- Idee: scan de boom niveau per niveau en voor elk niveau van links naar rechts.
- De wortel bevindt zich altijd bij index 0.
- Bij welke index bevindt zich het linker-kind van de knoop op index i ? $2i+1$
- Bij welke index bevindt zich het rechter-kind van de knoop op index i ? $2i+2$
- Bij welke index bevindt zich de ouder van de knoop op index i ? $(i-1)/2$ naar beneden afgerond
- Wat is het nadeel van een dergelijke array implementatie van bomen?

Inefficient geheugengebruik bij niet-complete bomen.

(Een default waarde moet afgesproken worden om lege cellen aan te duiden.)

➡ Een array implementatie wordt voornamelijk voor complete binaire bomen gebruikt.





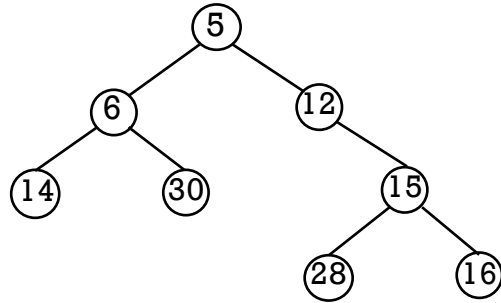
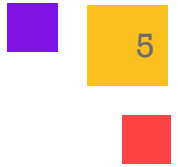
Heaps

Definitie

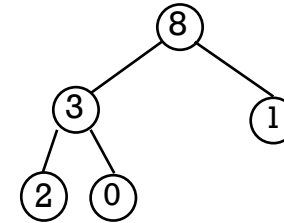
Een **heap** is een speciaal geval van een boom waarbij de data in de boom voldoen aan de heap eigenschap. Dit is 1 van volgende 2 eigenschappen:

- 1) *Voor elke knoop in de boom geldt dat zijn waarde groter of gelijk is aan de waarden van zijn kinderen. Men noemt dit een **max heap**.*
- 2) *Voor elke knoop in de boom geldt dat zijn waarde kleiner of gelijk is aan de waarden van zijn kinderen. Men noemt dit een **min heap**.*

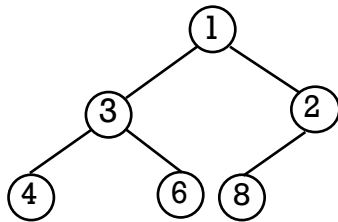
Voorbeelden



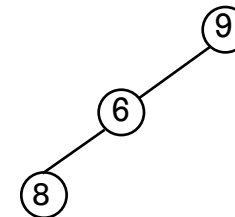
min heap



max heap



min heap



geen heap

Definitie

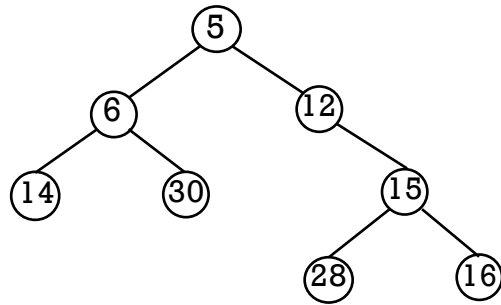
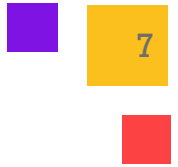
Een **heap** is een speciaal geval van een boom waarbij de data in de boom voldoen aan de heap eigenschap. Dit is 1 van volgende 2 eigenschappen:

- 1) *Voor elke knoop in de boom geldt dat zijn waarde groter of gelijk is aan de waarden van zijn kinderen. Men noemt dit een **max heap**.*
- 2) *Voor elke knoop in de boom geldt dat zijn waarde kleiner of gelijk is aan de waarden van zijn kinderen. Men noemt dit een **min heap**.*

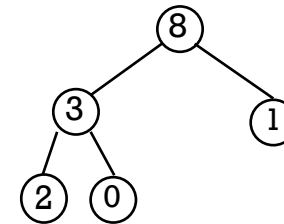
Een **binaire heap** is een complete binaire boom waarbij de data in de boom voldoen aan de heap eigenschap.

- 1) *Een **binaire max-heap** is een **complete binaire boom** waarbij voor elke knoop geldt dat zijn waarde groter of gelijk is aan de waarden van zijn kinderen.*
- 2) *Een **binaire min-heap** is een **complete binaire boom** waarbij voor elke knoop geldt dat zijn waarde kleiner of gelijk is aan de waarden van zijn kinderen.*

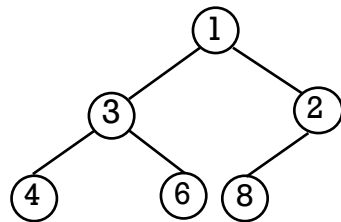
Voorbeelden



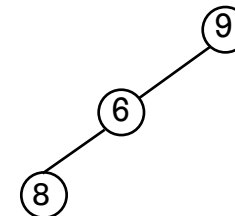
min heap
geen binaire heap



max heap
binaire max-heap



min heap
binaire min-heap



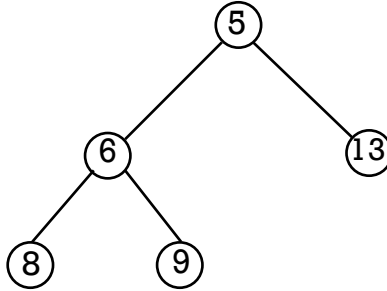
geen heap
geen binaire heap

Een waarde toevoegen aan een binaire min-heap

8

- 1) Voeg de waarde onderaan toe op de eerste vrije plaats.
- 2) Zolang de toegevoegde waarde kleiner is dan zijn ouder, verwissel de twee waarden.

Voorbeeld: voeg 4 toe aan onderstaande binaire min-heap.



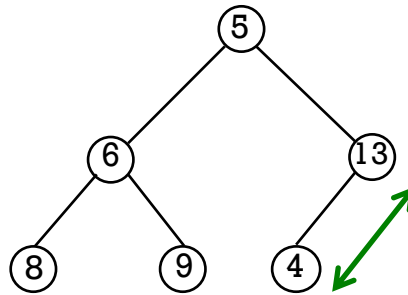
Zie: <https://visualgo.net/heap> opgelet dit is binaire max-heap

Een waarde toevoegen aan een binaire min-heap

9

- 1) Voeg de waarde onderaan toe op de eerste vrije plaats.
- 2) Zolang de toegevoegde waarde kleiner is dan zijn ouder, verwissel de twee waarden.

Voorbeeld: voeg 4 toe aan onderstaande binaire min-heap.

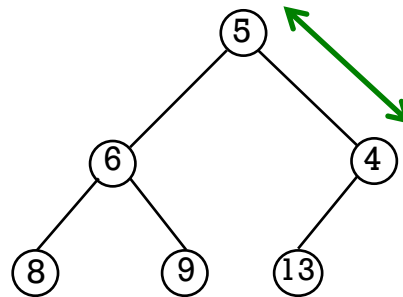


Een waarde toevoegen aan een binaire min-heap

10

- 1) Voeg de waarde onderaan toe op de eerste vrije plaats.
- 2) Zolang de toegevoegde waarde kleiner is dan zijn ouder, verwissel de twee waarden.

Voorbeeld: voeg 4 toe aan onderstaande binaire min-heap.

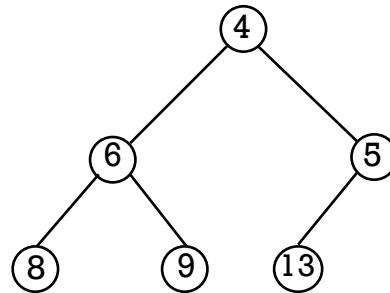


Een waarde toevoegen aan een binaire min-heap

11

- 1) Voeg de waarde onderaan toe op de eerste vrije plaats.
- 2) Zolang de toegevoegde waarde kleiner is dan zijn ouder, verwissel de twee waarden.

Voorbeeld: voeg 4 toe aan onderstaande binaire min-heap.



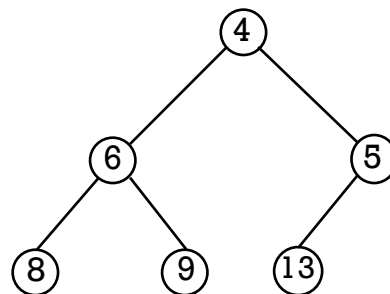
Deze operatie wordt ook wel **bubble-up** of **heapify-up** genoemd.

Delete het minimum in een binaire min-heap

12

- 1) Vervang de wortel met de waarde op de laatste plaats en delete de laatste knoop.
- 2) Zolang deze waarde groter is dan 1 van zijn kinderen, verwissel deze waarde met de kleinste waarde van zijn kinderen.

Voorbeeld: delete het minimum in onderstaande binaire min-heap.

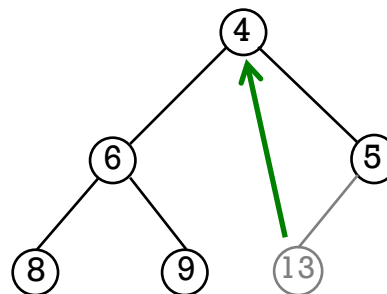


Delete het minimum in een binaire min-heap

13

- 1) Vervang de wortel met de waarde op de laatste plaats en delete de laatste knoop.
- 2) Zolang deze waarde groter is dan 1 van zijn kinderen, verwissel deze waarde met de kleinste waarde van zijn kinderen.

Voorbeeld: delete het minimum in onderstaande binaire min-heap.

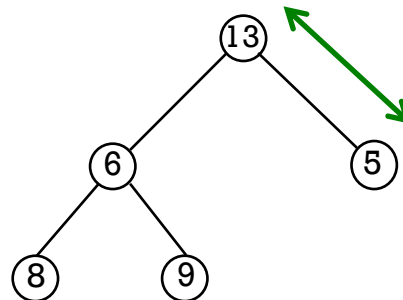


Delete het minimum in een binaire min-heap

14

- 1) Vervang de wortel met de waarde op de laatste plaats en delete de laatste knoop.
- 2) Zolang deze waarde groter is dan 1 van zijn kinderen, verwissel deze waarde met de kleinste waarde van zijn kinderen.

Voorbeeld: delete het minimum in onderstaande binaire min-heap.

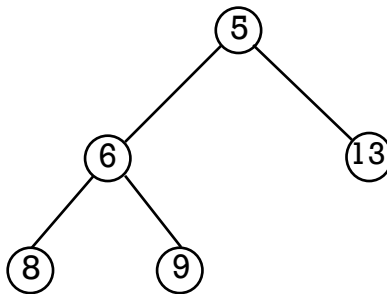


Delete het minimum in een binaire min-heap

15

- 1) Vervang de wortel met de waarde op de laatste plaats en delete de laatste knoop.
- 2) Zolang deze waarde groter is dan 1 van zijn kinderen, verwissel deze waarde met de kleinste waarde van zijn kinderen.

Voorbeeld: delete het minimum in onderstaande binaire min-heap.



Deze operatie wordt ook wel **bubble-down** of **heapify-down** genoemd.

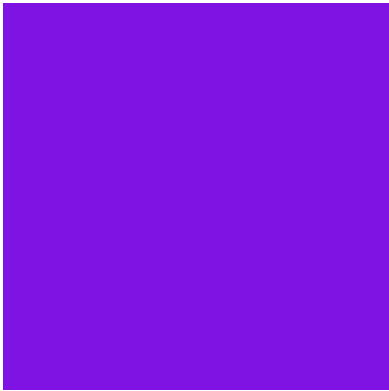
Implementatie van een binaire heap

16

Hoe ga je een binaire heap intern voorstellen?

Aangezien een binaire heap een complete binaire boom is, heeft een array implementatie de voorkeur.

Nut van een binaire heap?



Toepassing:
Priority queues

Priority queue

- Een queue is gebaseerd op het FIFO principe. Voorbeeld: een rij klanten die in een rij wachten op 1 of andere dienstverlening.
- Bij een priority queue wordt aan elke klant een prioriteit toegekend en wordt de klant met de hoogste prioriteit eerst bediend ongeacht de volgorde waarin de klanten aankwamen.
- Toepassing van priority queue: procesmanager van een OS
- Implementatie a.d.h.v. een lijst gesorteerd op prioriteit.
 - Nadeel: sorteren is een dure operatie terwijl alle elementen in de lijst niet moeten gesorteerd zijn!
- Een binaire heap blijkt een ideale keuze te zijn om de basisoperaties van een priority queue efficiënt te kunnen uitvoeren.

The image features three solid-colored squares arranged horizontally. On the left is a red square, in the center is a larger yellow square, and on the right is a purple square. The word 'Vragen?' is written in a dark, handwritten-style font in the center of the yellow square.

Vragen?