

Datastructuren

October 19, 2009

1 Heaps

1. Neem aan dat het array $A[1..N]$ gevuld is met gehele getallen. Er zijn twee voor de hand liggende manieren om dit array tot een heap om te zetten.

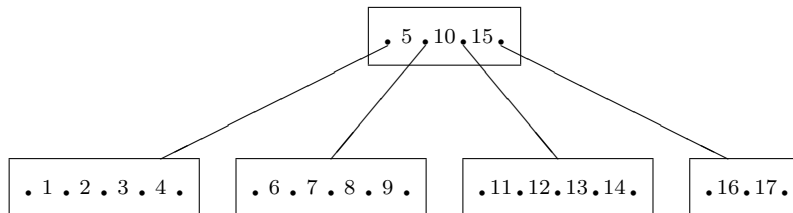
Bij de eerste manier werken we top-down: Als $A[1..i - 1]$ al een heap-structuur heeft, voegen we $A[i]$ toe en borrelen we dit element naar de juiste plaats. De tweede aanpak is bottom-up. We voegen telkens $A[i]$ samen met de twee heaps gevormd door de kinderen $A[2i]$ en $A[2i + 1]$ met al hun nakomelingen. Nu zakt $A[i]$ juist naar beneden.

- a Werk deze twee algoritmen verder uit tot pseudo-code of C++ functies.
 - b Bespreek de efficiëntie van de methoden. Welke manier heeft de voorkeur?
2. (\ddagger) Onderzoek hoe twee *heaps* gebruikt kunnen worden om van een verzameling snel zowel het grootste als het kleinste element te kunnen vinden (en verwijderen), terwijl nieuwe elementen in logaritmische tijd toegevoegd kunnen worden. Gebruik één van de heaps als *maxheap* (elke knoop heeft een sleutel die niet groter is dan die van zijn ouder) de ander als *minheap* (idem, maar sleutels niet kleiner dan die van ouder). Zorg ervoor dat tijdens het gebruik beide heaps ongeveer evenveel elementen blijven bevatten.

2 B-trees

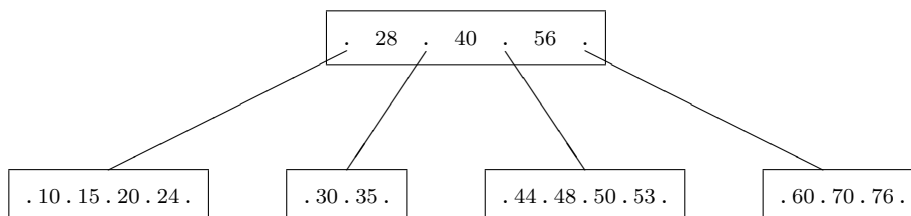
1. **a.** Bepaal de B-boom van orde 5 die ontstaat als
k, a, m, p, w, l, q, d, v, s, b, h, x, z, g, f, t, i, j, n, r, o
uit het gewone alfabet (met de normale ordening) in de aangegeven volgorde aan de aanvankelijk lege boom worden toegevoegd. De B-eigenschap moet steeds behouden blijven.
 - b** Welke boom ontstaat als eerst p en daarna m verwijderd wordt?
 - c** Als (a), maar nu een B-boom van orde 4, en met
w, p, l, m, v, a, x, d, s, e, r, t.
 - d** Welke boom ontstaat als uit deze boom r verwijderd wordt?
2. **a.** Wat is de minimaal mogelijke orde van een B-boom?

- b Hoeveel sleutels zijn er minimaal opgeslagen op het h -de niveau ($h \geq 1$) van een B-boom van orde 5 en hoogte $\geq h$? Idem voor maximaal.
- c Welke hoogte kan een B-boom van orde 5 hebben als daarin 490 sleutels zijn opgeborgen?
- d Geef een volgorde van de getallen 1, 2, ..., 16, 17 zodat bij het opbergen in een aanvankelijk lege B-boom van orde 5 de onderstaande boom ontstaat.



3. Dit was opgave 5 bij het tentamen Algoritmiëk van 17 augustus 1992.

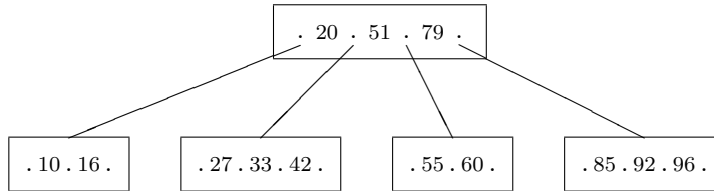
- a Hoeveel sleutels kan een knoop in een B-boom van orde 5 maximaal bevatten? En hoeveel minimaal?
- b Stel dat we een B-boom van orde 5 hebben met drie knopen. Hoeveel sleutels bevat hij dan minimaal? En hoeveel maximaal?
- c Gegeven de volgende B-boom van orde 5:



Welke boom ontstaat als we achtereenvolgens de getallen 25 en 45 toevoegen? De boom moet uiteraard steeds een B-boom van orde 5 blijven. Geef via tussenresultaten en begeleidende tekst duidelijk aan hoe u aan uw antwoord komt.

4. Dit was opgave 5 bij het tentamen Algoritmiëk van 11 juni 1993.

- a
- i. Hoeveel sleutels kan een knoop in een B-boom van orde 6 maximaal bevatten? En hoeveel minimaal?
 - ii. Hoe hoog kan een B-boom van orde 6 met zes knopen minimaal zijn? En hoe hoog maximaal? (Leg uit!)
 - iii. Hoeveel sleutels kan een B-boom van orde 6 met zes knopen minimaal bevatten? En hoeveel maximaal? (Leg uit!)
- b Gegeven de volgende B-boom van orde 6:



Welke boom ontstaat als we hieruit achtereenvolgens de getallen 51, 60 en 85 verwijderen? (Verwijderen dient te geschieden op de gebruikelijke wijze, dus via verwisselen met grootste kleinere, etc.!) De boom moet uiteraard steeds een B-boom van orde 6 blijven.

Geef duidelijk - via tussenresultaten en begeleidende tekst - aan hoe je aan je antwoord komt.

- c In C++ kan een B-boom van orde m (die verschillende gehele getallen bevat) als volgt gerepresenteerd worden:

```

struct BKnoop;
typedef BKnoop *BWijzer;
struct Element
{
    int Info;
    BWijzer Wijzer;
};
struct BKnoop
{
    int Aantal; // kan uiteindelijk alleen
               // waarden 1 ... m-1 aannemen
    BWijzer Vader, Links;
    Element Inhoud[m-1];
};
  
```

Zij een pointer Kind van type BWijzer gegeven, die wijst naar een knoop in een B-boom met daarin het minimale aantal sleutels dat een knoop mag bevatten. Zij gegeven dat Kind een rechterbuur heeft die meer dan het minimaal toegestane aantal sleutels bevat. Schrijf nu een stukje C++-tekst dat de meest rechtse sleutel uit de knoop waar Kind naar wijst, verwijdert en daarbij leent bij zijn rechterbuur. De betreffende knoop zit dus op het onderste niveau.

3 Self-Restructuring and Splay Trees

1. Consider the unbalanced binary search tree in Figure 3. In each of the following show what the tree will look like after key 'T' has been accessed.
 - a Self-restructuring tree with one rotation.
 - b Self-restructuring tree with rotate to the root.
 - c As a splay-tree.
 - d As a semisplaying tree.

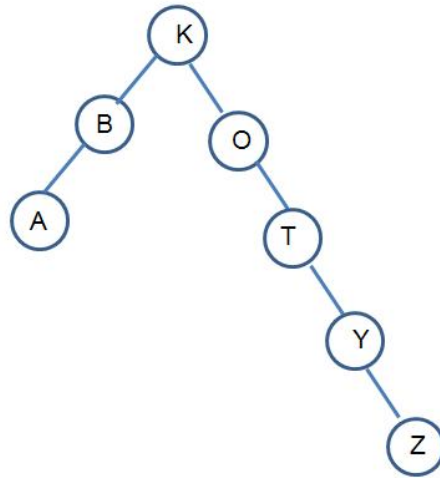


Figure 1: an unbalanced Binary Search Tree

4 AVL-trees

1. Deze opgave is afgeleid van opgave 4 uit het tentamen Algoritmiek van 3 juni 1992.
 - a Gegeven een binaire boom ter hoogte 5 (hoogte 1 correspondeert met een enkele knoop).
 - i. Hoeveel knopen bevat zo'n boom maximaal? En hoeveel minimaal? Geef van beide gevallen een voorbeeld.
Hoeveel verschillende binaire bomen met hoogte 5 zijn er met dat maximale, respectievelijk minimale aantal knopen?
 - ii. Hoeveel knopen kan zo'n boom maximaal, respectievelijk minimaal bevatten als deze tevens AVL-boom is?
Geef van beide gevallen een voorbeeld.
 - b Maak met behulp van plaatjes duidelijk wat een enkele rotatie naar rechts is. Schrijf vervolgens in C++ een stukje programma dat zo'n rotatie uitvoert bij de wortel van een binaire boom. Gebruik de volgende representatie van binaire bomen:

```

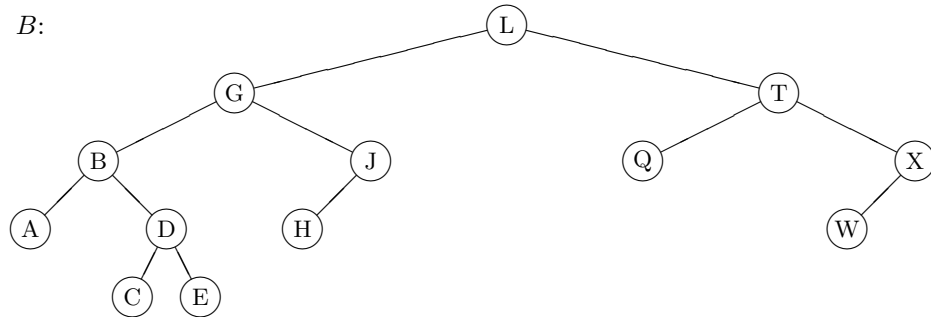
struct Knoop;
typedef Knoop *Wijzer;
struct Knoop
{
    char Info;
    Wijzer Links, Rechts;
};

```

- c Waar moet geroteerd worden om de AVL-structuur te herstellen als er met het gewone verwijderingsalgoritme voor zoekbomen (d.w.z.: verwijderen

van een knoop met twee kinderen gaat via verwisselen met de ‘grootste kleinere’) een sleutel is verwijderd?
 Hoeveel rotaties zijn er dan maximaal nodig?

d Gegeven de volgende AVL-boom B :



Verwijder met behulp van het gewone zoekboom-verwijderingsalgoritme uit B de sleutel T . Zorg er vervolgens voor (met behulp van rotaties) dat de boom een AVL-boom blijft. Geef via tussenresultaten en een korte toelichting duidelijk aan hoe u aan uw antwoord komt.

2. (‡) Geef bij alle antwoorden voldoende uitleg.

a De Fibonacci getallen worden gegeven door $F_0 = 0$, $F_1 = 1$ en $F_i = F_{i-1} + F_{i-2}$ voor $i \geq 2$. Er geldt $F_i \approx c\phi^i$, met $\phi = \frac{1}{2}(1 + \sqrt{5})$ en c een zekere constante waarde.

- i. Hoeveel knopen bevat een AVL-boom van hoogte $h \geq 1$ maximaal, en hoeveel minimaal (hierbij heeft de boom met 1 knoop hoogte $h = 1$)?
- ii. Leid hieruit een onder- en een bovengrens af voor de hoogte h van een AVL-boom met n knopen. Welke algemene conclusie kun je hieruit trekken?

b

- i. Hoeveel verschillende AVL-bomen van hoogte $h \geq 1$ met een maximaal aantal knopen zijn er?
- ii. Wat is het minimale aantal interne knopen in een AVL-boom van hoogte $h \geq 1$? En wat het minimale aantal bladeren?
- iii. Hoeveel verschillende AVL-bomen van hoogte $h \geq 1$ met een minimaal aantal knopen zijn er dus?

c

- i. Tot en met welk niveau is een AVL-boom van hoogte $h \geq 1$ in ieder geval in zijn geheel gevuld met knopen (de wortel bevindt zich op niveau 1)?
- ii. Leid hieruit een andere (weliswaar ruime) bovengrens af voor de hoogte h van een AVL-boom met n knopen.