

Optimale binaire zoekbomen

Project Algoritmen en Datastructuren II

Academiejaar 2013-2014

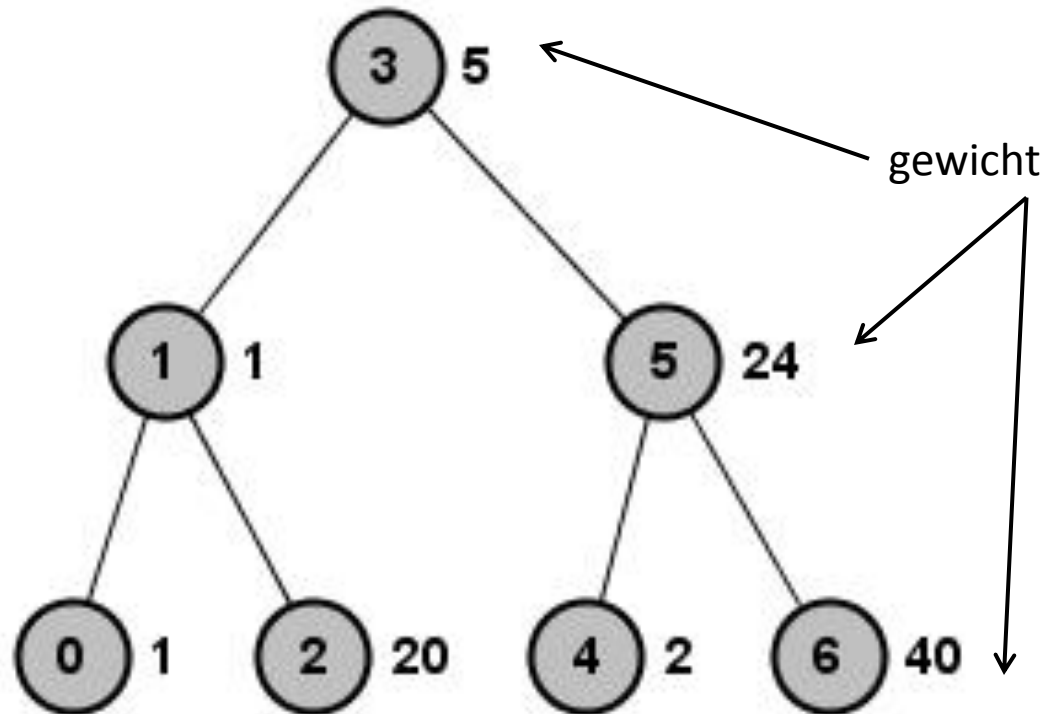


FACULTEIT WETENSCHAPPEN

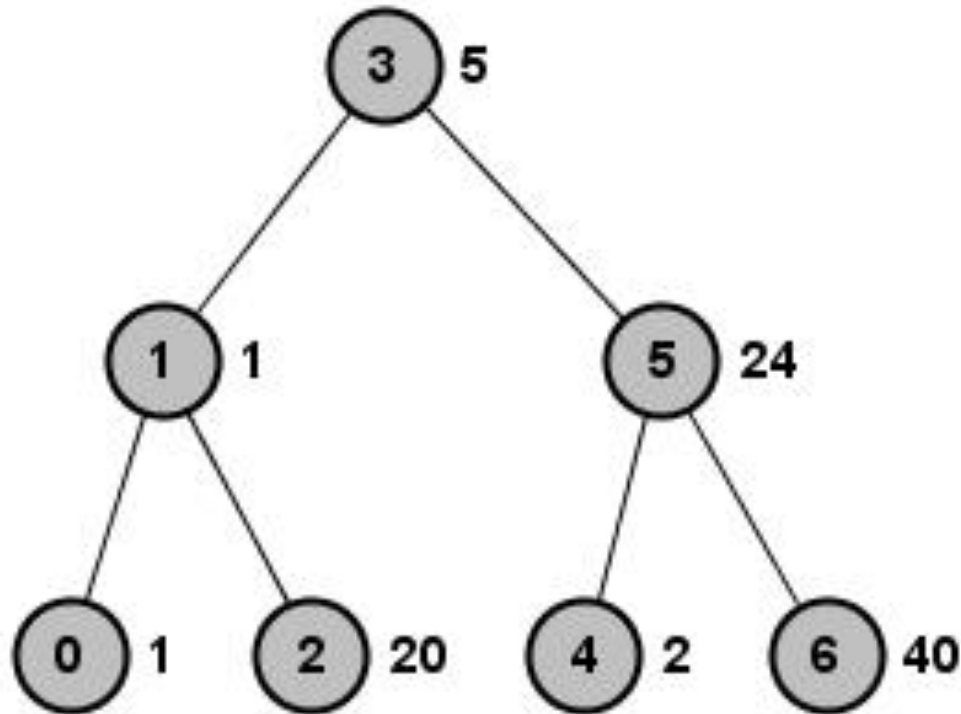
Doel van het project

- Implementeren van een dynamisch programmeren algoritme voor het bepalen van optimale binaire zoekbomen
- Implementeren van zoekbomen die automatisch geherorganiseerd worden: semi-splay bomen
- Grondig vergelijken van de methoden

Binaire zoekbomen: gewicht

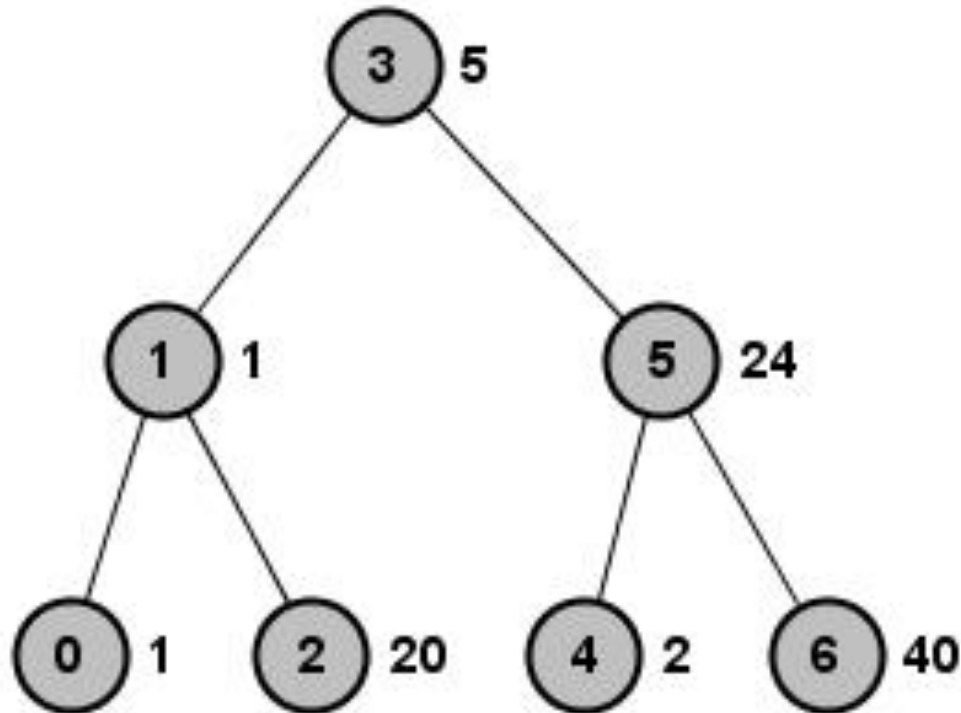


Kost opzoeken: navigaties



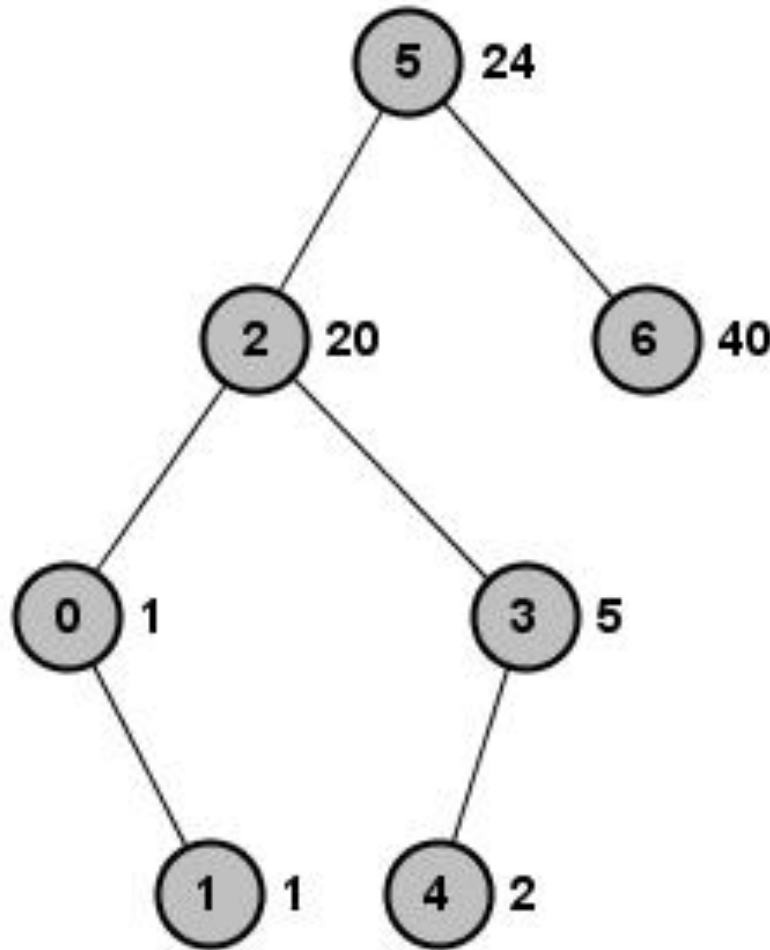
kost opzoeken sleutel = #navigaties = diepte

Kost zoekboom



Kost sleutel	=	gewicht · kost opzoeken sleutel
Kost boom	=	som van de kost van alle sleutels
Kost van deze boom	=	$5 \cdot 1 + 1 \cdot 2 + 24 \cdot 2 + 1 \cdot 3 + 20 \cdot 3 + 2 \cdot 3 + 40 \cdot 3$
	=	244

Optimale binaire zoekboom



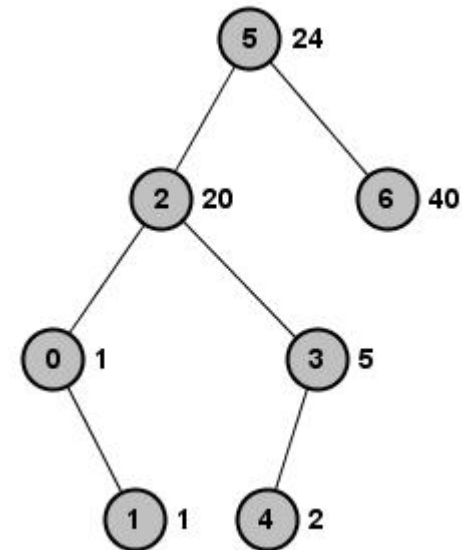
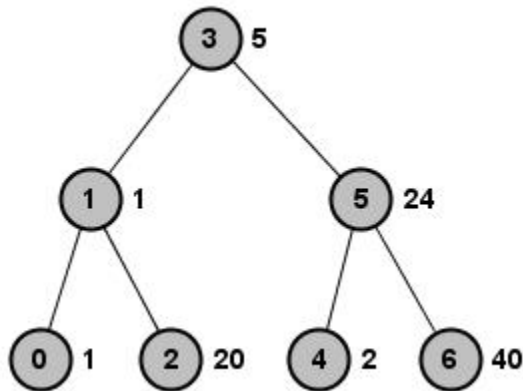
Kost = 174

Toepassing

- Bekomen van optimale zoekboom
- Gewichten niet op voorhand gekend!
- Herorganiseren van zoekboom
 - Verschillende mogelijkheden
 - Verschillende tijdstippen
- Implementatie en experimenten: Integers
- Andere toepassingen: objecten, records in databank, ...

Praktische vergelijking

Kleine bomen: geen verschil in uitvoeringstijd

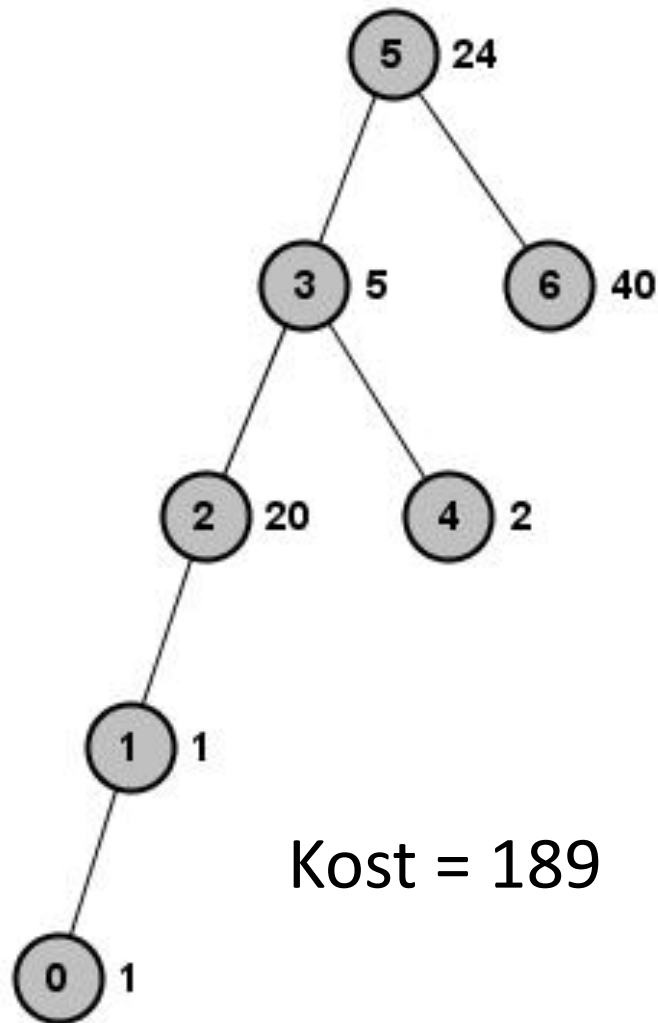


Praktische kost = uitvoeringstijd +
aantal navigaties · “kost van een navigatie”

Herorganiseren: optimale binaire zoekboom

- Recursief
 - Kost van een boom is de som van de kosten van de wortel, linker deelboom en rechter deelboom
 - Minimale kost van een boom hangt af van plaatsing wortel zodat de som minimaal is
- Dynamisch programmeren
 - Optimale kost en optimale binaire zoekboom
 - Top-down vs bottom-up

Herorganiseren: automatisch met semi-splay bomen



Implementatie

- Binaire zoekboom
 - add
 - contains
 - balance
- cost
- optimize
- Semi-splay bomen (add, contains)
- Bij onduidelijkheden: vragen!

Implementatie

- Alle binaire zoekbomen implementeren de interface `BST`
- Lees de opgave en het commentaar in de interface!
- Geef de klassen die binaire zoekbomen voorstellen de correcte naam.
- Zo efficiënt mogelijk...
- Duidelijke, leesbare code en commentaar

Vergelijking van de methoden

- Test de praktische aspecten van alle implementaties
 - Testdata kan worden aangemaakt via bijgeleverde programma's en worden ingelezen met de klasse `NumberReader`.
 - Maak **gevarieerde** en **voldoende grote** testdata aan.
 - Hoe duur of goedkoop een navigatie moet zijn om een methode voordeliger te maken?
- Vergelijk de theoretische aspecten van de methoden
- Test ook de correctheid van de algoritmen!

Verslag

- Lees de opgave goed en antwoord op alle vragen!!!
- Bespreek gebruikte datastructuren en algoritmen
- Licht ontwerpbeslissingen toe
- Beschrijf je tests
- Vergelijk de verschillende methodes
 - Gebruik overzichtelijke en correcte grafieken en tabellen.
- Verklaar de resultaten
- Antwoord op de theoretische vragen
- Vorm duidelijke conclusies!
- ...



Indienen

- Elektronisch via <http://indiano.ugent.be>
- Tussentijdse versie: **vrijdag 1 november 2013 om 17u00** (werkende implementatie van een binaire zoekboom en dynamisch programmeren algoritme)
- Definitieve versie: **maandag 25 november 2013 om 17u00**
 - Zorg voor een **papieren** versie van het verslag
- Tussentijds indienen is verplicht

Algemene richtlijnen

- Project op **4** van de 20 punten
 - Punten worden overgenomen naar tweede zit
- Ten strengste verboden om code uit te wisselen of van het internet te kopiëren
- Essentiële vragen worden **niet** meer beantwoord tijdens de laatste week voor de deadline.