

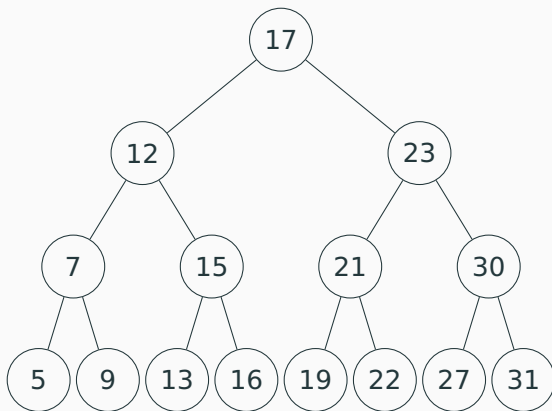
Binary Search Tree (BST)

Programmering-B

Thomas Birk Abildgaard

February 25, 2023

HTX Frederikshavn



Binary Search Trees

(Binaere Søgetræer)

Binary Search Tree (BST)

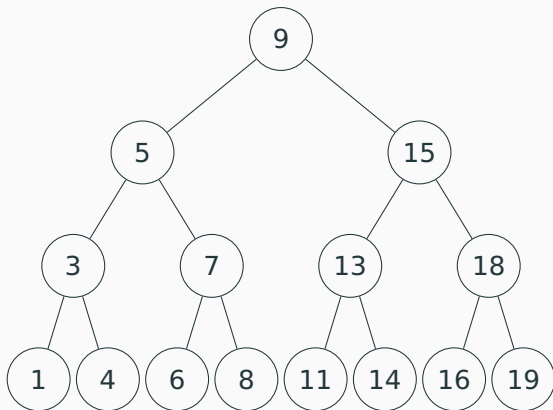
- Træstruktur (rekursiv)
 - Simpel struktur
 - Hurtig opslag/søgning
- Består af nodes (knuder)
- Hver node indeholder en værdi (eg. int)
- BST'et starter med en root node (rod knude)
- Hver node kan have to child nodes (børn)
 - **left** node (kan være null, for "leaf" node.)
 - **right** node (kan være null, for "leaf" node.)

- Første node er root node
- node med "mindre" værdi end root node indsættes som left-child-node (venstre)
- node med "større" værdi end root node indsættes som right-child-node (højre)
- Eksisterer der iforvejen en left/right node, betragtes denne (rekursivt) som root node, og der sammenlignes igen som ovenstående.

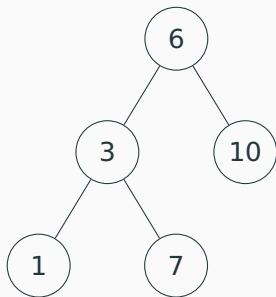
- Første node er root node
- node med "mindre" værdi end root node indsættes som left-child-node (venstre)
- node med "større" værdi end root node indsættes som right-child-node (højre)
- Eksisterer der iforvejen en left/right node, betragtes denne (rekursivt) som root node, og der sammenlignes igen som ovenstående.

hmn.. basis-tilfælde ?, rekursivt tilfælde ?

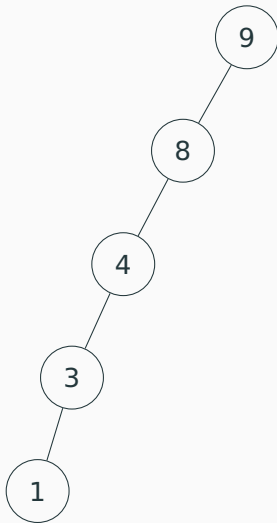
BST en rekursiv datastruktur



Korrekt BST ?

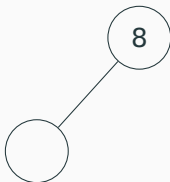


Korrekt BST ?



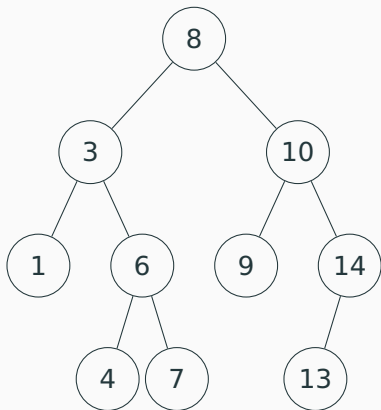
Konstruktion af BST #1

8, 3, 10, 1, 6, 14, 4, 13, 9

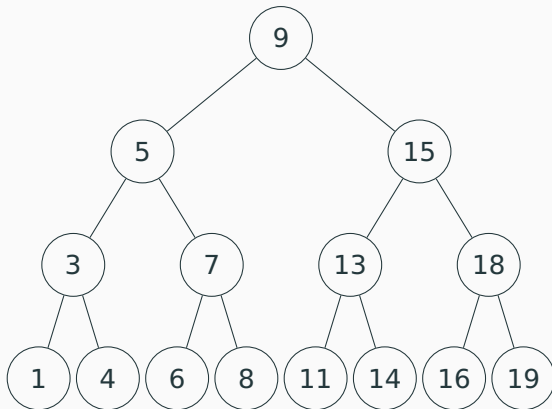


Konstruktion af BST #2

8, 3, 10, 1, 6, 14, 4, 13, 9



BST indsæt værdien 17



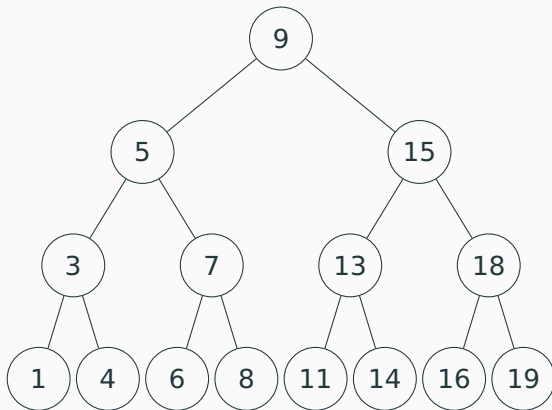
Øvelse : Analog opbygning af BST

Søgning i et BST

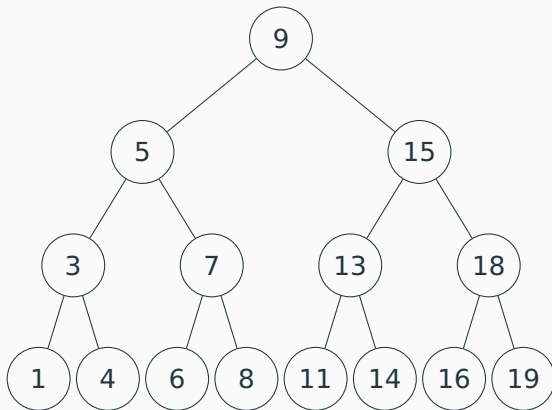
Search **key** (søgenøgle)

- Hvis søgetræet er tomt, blev data ikke fundet
- Hvis data i den aktuelle knude er lig med **key** er data fundet
- Hvis **key** er mindre end data i den aktuelle knude, så gentag søgningen i det **venstre** undertræ
- Hvis **key** er større end data i den aktuelle knude, så gentag søgningen i det **højre** undertræ

Søg efter værdien 16

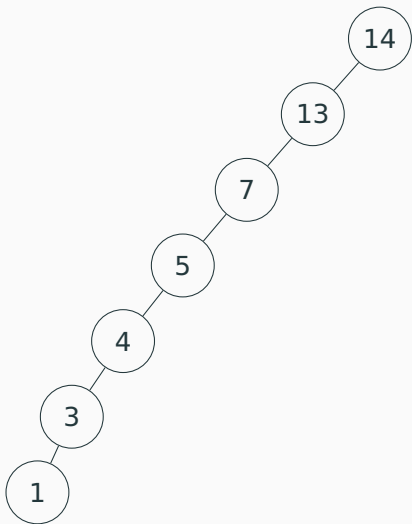


Søg efter værdien 16



9, 5, 15, 3, 1, 7, 8, 6, 18, 13, 15, 11, 19, **16**

Men.. hvad nu hvis



14, 13, 7, 5, 4, 3, **1**

Øvelse

1. Find fejlen(e) i eksisterende BST's
2. Konstruktion af BST fra tæl række
3. Pseudokode for opbygning af BST. (ei. hvordan indsætter man en node)

- Rekursiv datastruktur
- BST (Binary Search Tree)
 - Node & Leaf
 - Root node
 - Pseudokode (konstruktion af træ)
- Rekursion
 - Basis-tilfælde
 - Rekursivt-tilfælde
- Søgning i BST (un poco)
 - Balanceret vs. Ubalanceret

**Hvad har BST med programmering at
gøre ?**

This page is intentionally left blank

Øvelse : BST nodes i C#

Implementer en node (knude) i C#.

dvs. Konstruer en klasse "Node" ell. "Knude" der indeholder en integer værdi, samt en **left** og **right** reference.

- Experimenter med opretning & sammensætning af knuder til træ-konstruktion.
- Experimenter med udskrivning af træ-konstruktionen til consol. (el. evt. anden visualisering)

Vi følger op på BST operationer når vi mødes igen næste gang.