

# 3

## Binaire zoekbomen BST

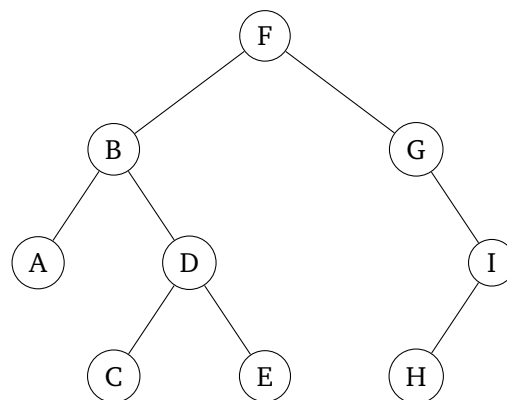
### Inleiding

Download het bestand BG\_Les4.zip van Toledo. Importeer dit bestand in Eclipse als volgt:

```
File > Import... > General > Existing Projects into Workspace  
> Next > Select archive file > Finish
```

### Oefening 3.1

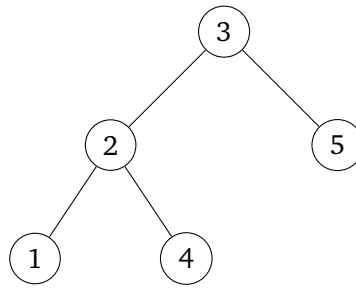
Gegeven twee binaire bomen in figuren 3.1 en 3.2. Ga voor elk van beide na of het een binaire zoekboom is (BST).



**Figuur 3.1** Een binaire boom, maar is het ook een BST?

### Oefening 3.2

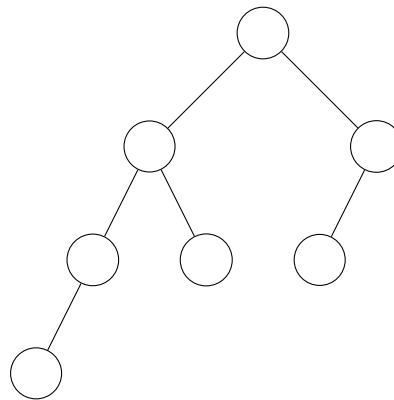
Kan je een BST doorlopen met 1 van de 3 strategieën besproken in les 3 (pre-order, in-order of post-order) zodanig dat de knopen worden bezocht van klein naar groot?



**Figuur 3.2** Een binare boom met getallen in de knopen

### Oefening 3.3

Op hoeveel verschillende manieren kan je de getallen 3 tot en met 9 in onderstaande knopen invullen met als resultaat een BST? Teken deze verschillende mogelijkheden.



**Figuur 3.3** Getallen van 3 t.e.m. 9 invullen in de knopen

### Oefening 3.4

Bestudeer de klasse `BinaryTree` in de package `domain` in de `src` folder. Een gebruiker kan enkel operaties uitvoeren op een BST via enkele publieke methoden. Een eerste dergelijke methode is `lookup` waarmee snel een bepaalde waarde in de BST kan opgezocht worden (zie de slides van deze les). Een tweede methode is `addNode` waarmee gegeven data aan de BST kan toegevoegd worden.

- Teken op papier eerst de binaire zoekboom zoals die door de `main` methode in de klasse `BinarySearchTreeDriver` zal worden toegevoegd.
- Implementeer de `addNode` methode in de `BinarySearchTree` klasse.
- Om je implementatie te controleren run je de `BinarySearchDriver` klasse uit de `ui` package. Verwachte uitvoer: 3 4 5 6 7 8 9

### Oefening 3.5

Het doel van deze oefening is een implementatie te maken van een methode die de grootste waarde uit de BST teruggeeft.

- a) Implementeer de `searchGreatest` methode in de `BinarySearchTree` klasse.
- b) Om je implementatie te controleren run je de `BinarySearchDriver` klasse uit de `ui` package. Verwachte uitvoer: 9

### Oefening 3.6

Het doel van deze oefening is een implementatie te maken van een methode die de kleinste waarde uit de BST teruggeeft.

- a) Implementeer de `searchSmallest` methode in de `BinarySearchTree` klasse.
- b) Om je implementatie te controleren run je de `BinarySearchDriver` klasse uit de `ui` package. Verwachte uitvoer: 3

### Oefening 3.7

Een volgende methode is `removeNode` waarmee gegeven data uit de BST zal verwijderd worden indien mogelijk. Eerst breiden we de boom een beetje uit met twee extra knopen.

- a) Breid in de `main` methode van de klasse `BinarySearchTreeDriver` de gegeven boom uit met twee extra datavelden: 10 en 11. Teken de nieuwe boom.
- b) Implementeer de `removeNode` methode in de `BinarySearchTree` klasse.
- c) Verwijder de knoop met data-veld 9.

### Oefening 3.8

Een probleem dat weggesnoeid moet worden ...

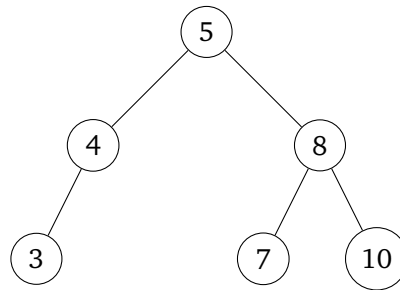
- a) Gebruik de `countNodes()` methode die je in vorige les maakte om het aantal knopen voor en na de verwijdering van de knoop met data-veld 9 te tellen. Je merkt ongetwijfeld iets eigenaardigs op: het verwijderen van de knoop heeft geen effect op het aantal knopen. We onderzoeken dat in de volgende deelvragen.
- b) Teken de boom die je na het verwijderen van de knoop met waarde 9 kreeg. Eén van de bladeren van de boom heeft nu als data de waarde `null` gekregen. Met de in-orderwandeling kan je deze bladeren niet visualiseren. Eventueel maak je een alternatieve `printInOrderNull()` methode die de string "null" print als het dataveld als waarde `null` heeft.

### 3 Binaire zoekbomen BST

- c) Blaadjes met als datawaarde `null` laten we niet aan de boom staan. Tijd om te snoeien! Schrijf een methode `ruimOp()` die deze ‘verdorde’ blaadjes verwijdert. Pas ze toe op de boom na het verwijderen van de knoop met waarde 9 en laat zien dat het aantal knopen na de snoeibeurt wel degelijk ééntje verminderd is.
- d) Verwijder tenslotte uit de boom data-veld 11 en data-veld 6 en ruim lege blaadjes op. De verwachte uitvoer van een in-order wandeling is dan: 3 4 5 7 8 10.

#### Oefening 3.9

De bedoeling van deze oefening is een methode te schrijven die gegeven een data-veld een pad teruggeeft van de wortel van de boom tot het dataveld indien mogelijk. We passen dit toe op de boom die je als resultaat na vorige oefening zou moeten bekomen. Bij wijze van controle: figuur 3.4 toont deze boom met zes knopen.



**Figuur 3.4** BST na uitbreiding met knopen 10 en 11 en verwijdering van 9, 11 en 6

- a) Implementeer de `getPath` methode in de `BinarySearchTree` klasse.
- b) Als controle pas je de klasse `BinarySearchTreeDriver` aan zodat je drie keer de methode `getPath` oproept met als parameter respectievelijk 7, 4 en 8. De uitvoer moet dan respectievelijk zijn: bij 7: [5,8,7] ; bij 4: [5, 4] en bij 8: [5, 8]. Probeer de methode ook uit met als parameter 22. In dit geval moet de methode `null` als returnwaarde hebben.