

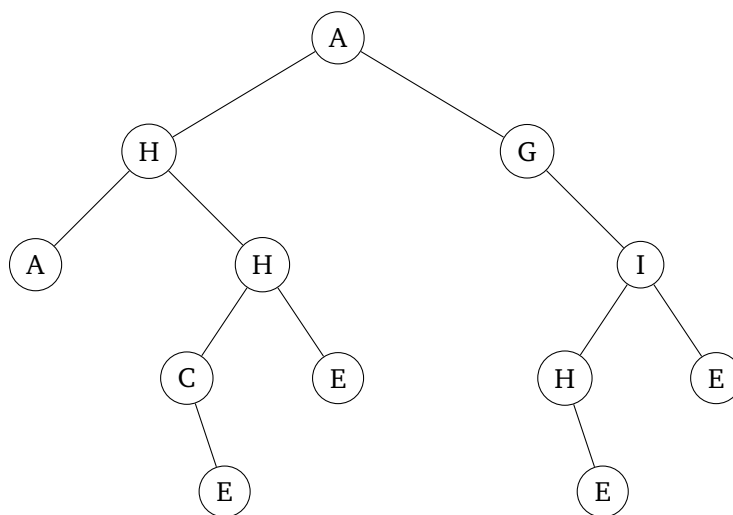
5

Herhalings oefeningen binaire bomen

Oefening 5.1

Gegeven de implementatie van een binaire boom zoals in les 3 gegeven. De bedoeling van deze oefening is het aantal voorkomens van een gegeven data-veld in een binaire boom te tellen.

- Implementeer een methode `count` in de `BinaryTree` klasse die het aantal voorkomens van een gegeven data-veld telt in een boom.
- Maak een nieuwe `BinaryTreeDriver` klasse waarin je de boom overeenkomstig figuur 5.1 maakt.



Figuur 5.1 Binaire boom met herhaalde datavelden

- Run de `main` functie in de `BinaryTreeDriver` klasse en controleer hiermee de implementatie van de `count` methode met respectievelijke parameters: I, A, H, E en Q. Verwachte uitvoer:
Aantal voorkomen van I = 1
Aantal voorkomens van A = 2
Aantal voorkomens van H = 3
Aantal voorkomens van E = 4

5 Herhalingsoefeningen binaire bomen

Aantal voorkomens van Q = 0

Oefening 5.2 a) Schrijf een methode `getNodesAtDistance(k)` in de `BinaryTree` klasse die een lijst teruggeeft van de datavelden die op een afstand k van de wortel van de binaire boom verwijderd zijn. In de boom uit figuur 5.1 zijn A, H en I de datavelden van knopen die op een afstand 2 van de root verwijderd zijn.

b) Test je implementatie uit door in de klasse `BBDriver` de methode `getNodesAtDistance` op te roepen met parameters 0, 1, 2, 3 en 4. Verwachte uitvoer:

Datavelden van knopen verwijderd op een afstand van 0 van de root = [A]

Datavelden van knopen verwijderd op een afstand van 1 van de root = [H, G]

Datavelden van knopen verwijderd op een afstand van 2 van de root = [A, H, I]

Datavelden van knopen verwijderd op een afstand van 3 van de root = [C, E, H, E]

Datavelden van knopen verwijderd op een afstand van 4 van de root = [E, E]

Oefening 5.3

Gegeven volgende code:

Listing 1 Mystery methode

```
public ArrayList<E> mystery() {
    ArrayList<E> lijst = new ArrayList<>();
    if (this.leftTree != null) lijst.add(this.leftTree.data);
    if (this.rightTree != null) lijst.add(this.rightTree.data);
    return lijst;
}

public ArrayList<E> mystery(int g) {
    if (g == 1) {
        return this.mystery();
    } else {
        ArrayList<E> links = new ArrayList<>();
        if (this.leftTree != null) links = this.leftTree.mystery(g - 1);
        ArrayList<E> rechts = new ArrayList<>();
        if (this.rightTree != null) rechts = this.rightTree.mystery(g - 1);
        links.addAll(rechts);
        return links;
    }
}
```

Geef nu de uitvoer van volgende oproepen van deze methode voor de boom uit figuur 5.1:

a) `boom.mystery(1)`

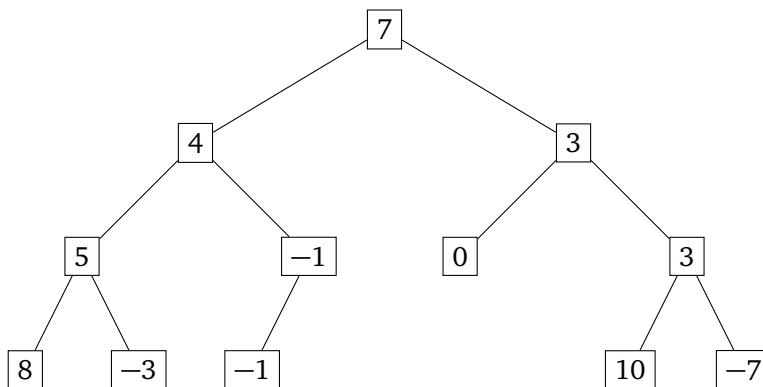
b) `boom.mystery(2)`

- c) `boom.mystery(3)`
- d) `boom.mystery(4)`
- e) `boom.mystery(5)`

Oefening 5.4

Schrijf een methode `kinderSom()`. Het resultaat is een boolean. De methode geeft `true` terug als de boom waarop ze toegepast wordt voldoet aan de eigenschap dat elke knoop als waarde de som van zijn kinderen heeft (dat aantal kinderen kan natuurlijk 2, 1 of 0 zijn). Figuur 5.2 toont een boom die aan deze eigenschap voldoet.

Je zal voor deze methode een nieuwe klasse moeten schrijven. Je kan immers niet zomaar de waarde van twee knopen van een nog te bepalen type `E` optellen, bvb. als je voor `E` het type `String` kiest! Beperk je dus voor deze oefening tot een nieuw soort binaire boom, nl. één waarbij het veld `data` een `int` is. Test je methode uit door de boom van figuur 5.2 te implementeren in een `main`methode en het resultaat van de toepassing van deze methode op deze boom naar de console te schrijven.



Figuur 5.2 Binaire boom voldoet aan `kinderSom`

6

Herhalingsoefeningen BST

Oefening 6.1

De preorder methode van een BST geeft volgende uitvoer: 30, 20, 10, 15, 25, 23, 39, 35, 42. Geef de postorder uitvoer.

Oefening 6.2

We wensen een BST te bouwen bestaande uit de gehele getallen 1 tot en met 10. In welke volgorde moeten de getallen worden toegevoegd opdat de resulterende boom compleet is?

Oefening 6.3

Stel dat 7, 5, 1, 8, 3, 6, 0, 9, 4 en 2 worden toegevoegd aan een lege BST. Wat is de diepte van de resulterende boom?

Oefening 6.4

Stel dat we een BST maken door vertrekkende van een lege BST vervolgens de getallen 71, 65, 84, 69, 67 en 83 toe te voegen. Welke zijn de data-velden van de bladeren van de boom?

Oefening 6.5

Stel dat boom een BST is bestaande uit gehele getallen groter dan of gelijk aan 1 en kleiner dan of gelijk aan 100. Welke van de volgende paden kan (kunnen) niet?

- a) 10, 75, 64, 43, 60, 57, 55
- b) 90, 12, 68, 34, 62, 45, 55
- c) 9, 85, 47, 68, 43, 57, 55
- d) 79, 14, 72, 56, 16, 53, 55