

Aanzet tot oplossingen les 3

Bomen en Grafen

Binaire Bomen

Oefening 2.5: Aantal Knopen

Basisgeval

lege boom
aantalKnopen = 0

```
als (this.data == null) {  
    aantalKnopen = 0;  
}
```

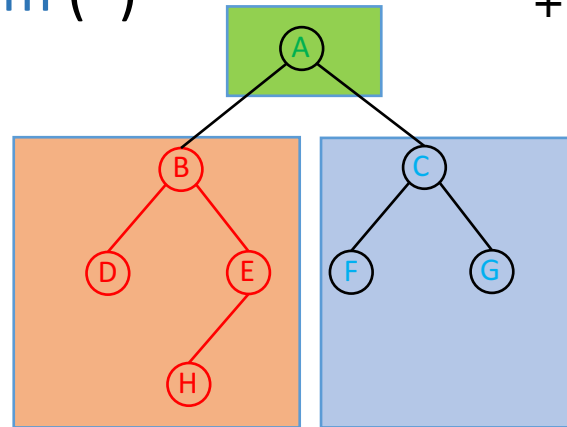
Algemene regel

aantalKnopen =

de wortel

+ aantalKnopen Linkerboom (*)

+ aantalKnopen Rechterboom (*)



(*) indien ze bestaan

aantalKnopen =

1

+ aantalKnopen Linkerboom (*)

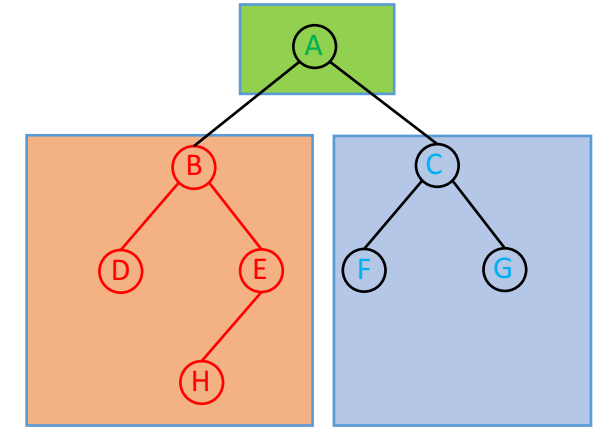
+ aantalKnopen Rechterboom (*)

Berekenen met recursieve oproep
van de methode

(*) if (leftTree / rightTree != null)

Pseudo-code

```
if (basisgeval)
    return 0;
else {
    int aantalKnopen = 1;
    if (linkerboom bestaat)
        aantalKnopen += linkerBoom.aantalKnopen;
    if (rechterboom bestaat)
        aantalKnopen += rechterBoom.aantalKnopen;
    return aantalKnopen;
}
```



Oefening 2.6: Maximale Diepte

Basisgeval

lege boom
maximaleDiepte= 0

```
als (this.data == null) {  
    maximaleDiepte= 0;  
}
```

Algemene regel

maximaleDiepte =

de wortel

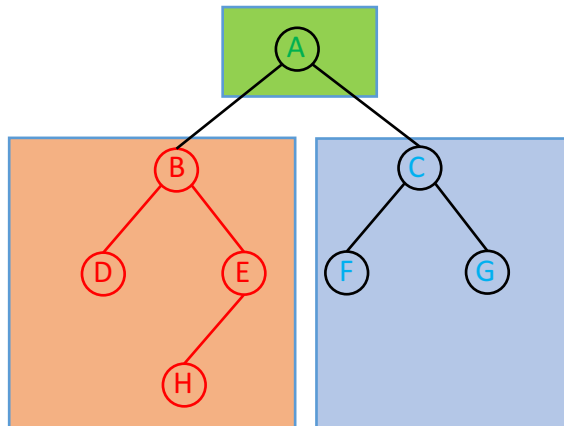
+ maximum van

maximaleDiepte Linkerboom (*)

en

maximaleDiepte Rechterboom (*)

(*) indien ze bestaan



maximaleDiepte =

1

+ max(

maximaleDiepte Linkerboom (*)

, maximaleDiepte Rechterboom (*)
)

Berekenen met recursieve oproep
van de methode

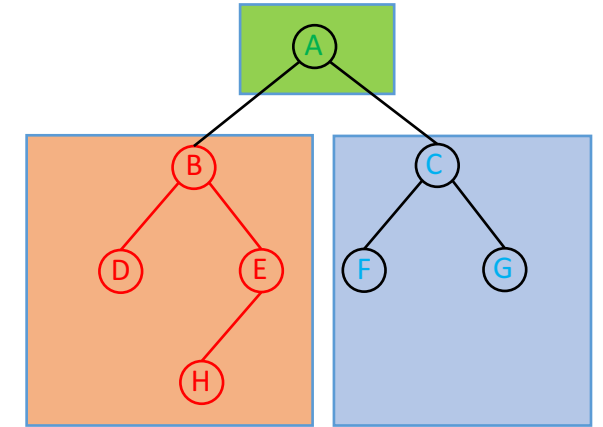
(*) if (leftTree / rightTree != null)

Pseudo-code

```
if (basisgeval)
    return 0;

else {
    int maximaleDiepteLinks = 0;
    if (linkerboom bestaat)
        maximaleDiepteLinks = linkerBoom.maximaleDiepte;
    int maximaleDiepteRechts = 0;
    if (rechterboom bestaat)
        maximaleDiepteRechts = rechterBoom.maximaleDiepte;

    return 1 + max(maximaleDiepteLinks, maximaleDiepteRechts);
}
```



Defining 2.10: Contains

Basisgeval

lege boom
antwoord = false

```
als (this.data == null) {  
    antwoord = false;  
}
```

Algemene regel

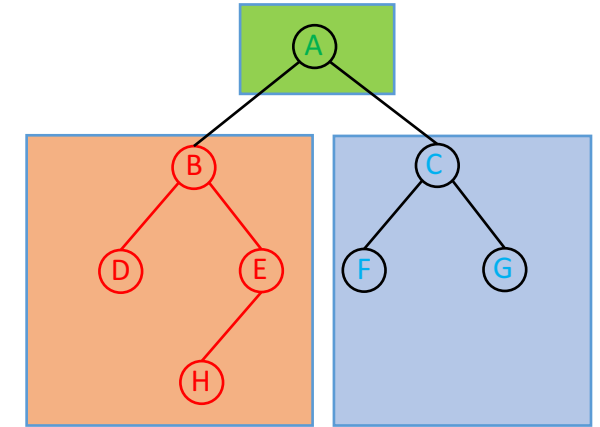
Als **wortel** == gezochteKnoop:

- gevonden

Anders

- zoek in **linkerBoom** (*)
- zoek in **rechterBoom** (*)
- als in één van beide gezochteKnoop aanwezig: gevonden

(*) indien die bestaat

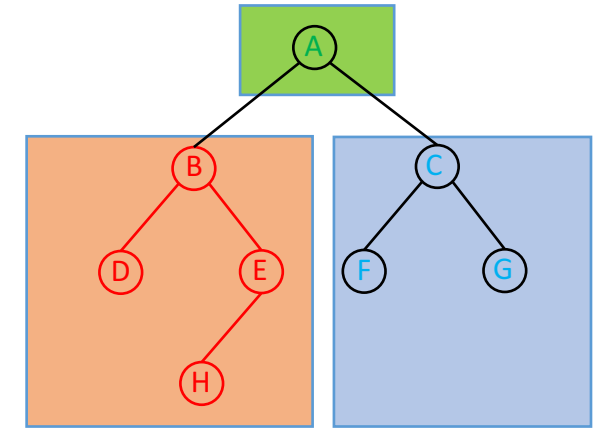


Pseudo-code

```
if (basisgeval)
    return false;

else {
    if (wortel == gezochteKnoop)
        return true;
    boolean gevondenLinks = false;
    if (linkerboom bestaat)
        gevondenLinks = linkerBoom.contains();
    boolean gevondenRechts = false;
    if (rechterboom bestaat)
        gevondenRechts = rechterBoom.contains();

    return gevondenLinks || gevondenRechts;
}
```



Simulatie: zoek knoop E

