

Vakdidactiek Informatica

Binaire Bomen

Sara Van Langenhove

svlangen@cage.UGent.be

AILO Informatica

1 December 2003

Herhaling

Top-Down methode

- probleem opsplitsen in steeds kleiner wordende deelproblemen

Algoritme

- opeenvolging van stappen die leiden tot oplossing

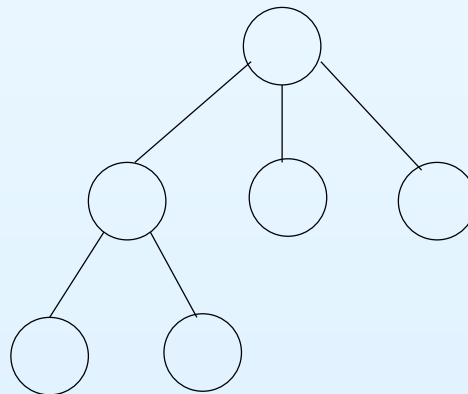
Recursie

- procedure roept zichzelf op
- probleem oplossen door reductie naar analoge en kleinere deelproblemen

Bomen

Boom

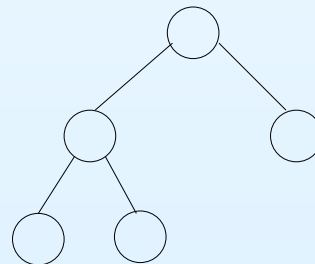
- Basisdefinitie
 - is een lege boom
 - of is een knoop (wortel) bestaande uit een **willekeurig aantal** deelbomen
- Terminologie
 - wortel : topknoop van de boom
 - takken : verbindingen tussen de knopen → paden
 - bladeren : knopen met lege deelbomen
 - hoogte : padlengte tussen knoop en zijn diepste blad



Binaire Bomen

Binaire Boom

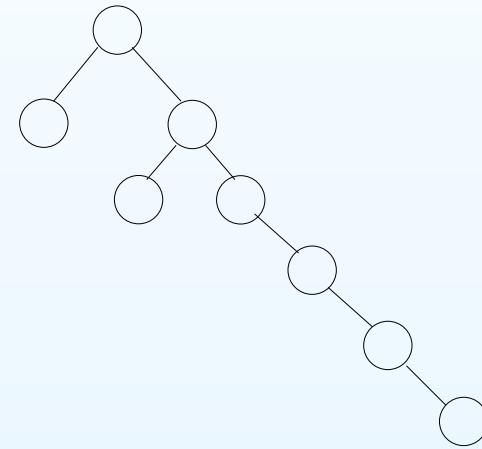
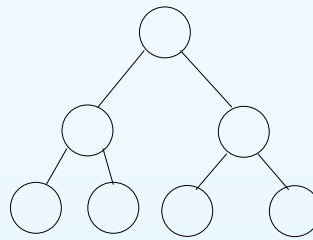
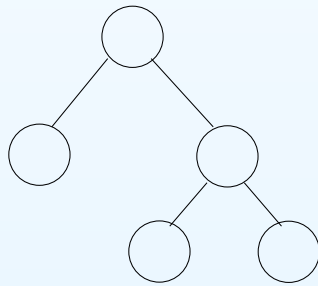
- Basisdefinitie
 - is een lege boom
 - of is een knoop (wortel) bestaande uit **maximaal twee** deelbomen
 - linkerdeelboom → binaire boom
 - rechterdeelboom → binaire boom
- Terminologie
 - wortel : topknoop van de boom
 - takken : verbindingen tussen de knopen → paden
 - bladeren : knopen met lege deelbomen
 - hoogte : padlengte tussen knoop en zijn diepste blad



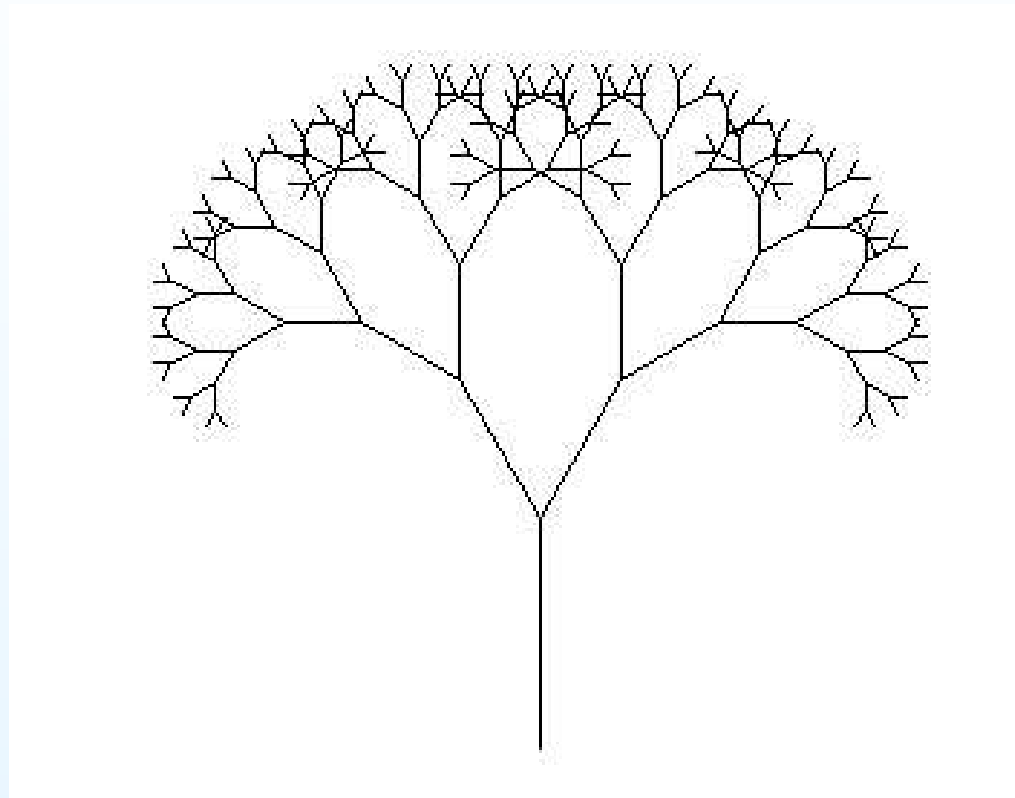
Gebalanceerde Binaire Bomen

Gebalanceerde Binaire Boom

- een binaire boom
- \forall knoop: verschil in hoogte linker- en rechterdeelboom **max. 1**



Binaire Bomen in Logo



Tips

- identificeer kleinere deelproblemen
- bepaal algoritme
- recursie!

Binaire Bomen in Logo - Top Down

Probleem opsplitsen in deelproblemen (top-down)

- Binaire boom
 - bestaat uit een stam
 - Op stam staan 2 kleinere binaire bomen (links en rechts)
- Linkse binaire deelboom
 - bestaat uit een stam
 - Op stam staan 2 kleinere binaire bomen (links en rechts)
- Rechtse binaire deelboom - analoog
- ...

Binaire Bomen in Logo - Algoritme

```
tree:
  empty
  stam linkerdeelboom rechterdeelboom
empty:
  = stopvoorwaarde
stam:
  lijnstuk tekenen
linkerdeelboom:
  tree
rechterdeelboom:
  tree
```


Binaire Bomen in Logo - Algoritme

```
tree:
  empty
  stam linkerdeelboom rechterdeelboom
empty:
  = stopvoorwaarde
stam:
  lijnstuk tekenen
linkerdeelboom:
  tree
rechterdeelboom:
  tree
```

```
to tree :SIZE
  if :SIZE < 5 [STOP] ; boom is opgebouwd
  FD :SIZE             ; tekenen van de stam
  LT 30                ; positioneren voor linkerdeelboom
  tree :SIZE*.7        ; plaatsen van de linkerdeelboom
  RT 60                ; positioneren voor rechterdeelboom
  tree :SIZE*.7        ; plaatsen van de rechterdeelboom
end
```

Binaire Bomen in Logo - Fout?

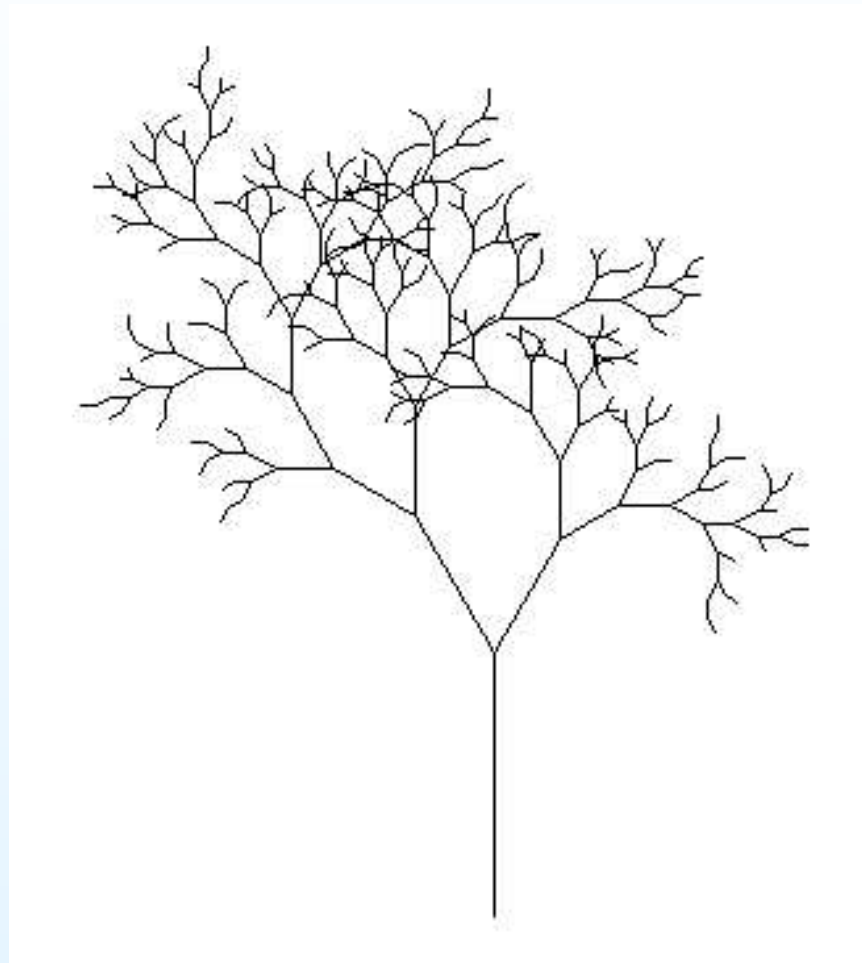
Oorzaak fout?

- veronderstelling
 - tekenen deelbomen vanuit juiste positie en juiste richting
- aanpassing
 - na tekenen van lege deelbomen → terugkeren
 - zorgen voor juiste positie en juiste richting

Binaire Bomen in Logo - Fout?

```
tree 9
  if 9 < 5 [STOP]
  FD 9
  LT 30
  tree 6      ;LinkerDeelBoom
    if 6 < 5 [STOP]
    FD 6
    LT 30
    tree 4    ;Lege LinkerDeelboom
      if 4 < 5 [STOP]
    RT 60
    tree 4    ;Lege RechterDeelBoom
      if 4 < 5 [STOP]
  RT 60      ;LinkerDeelBoom afgewerkt, begin met RechterDeelBoom
  tree 6      ;RechterDeelBoom
  ....
```

Binaire Bomen in Logo - Beter Uitzicht



Hoe?

Binaire Bomen in Logo - Beter Uitzicht

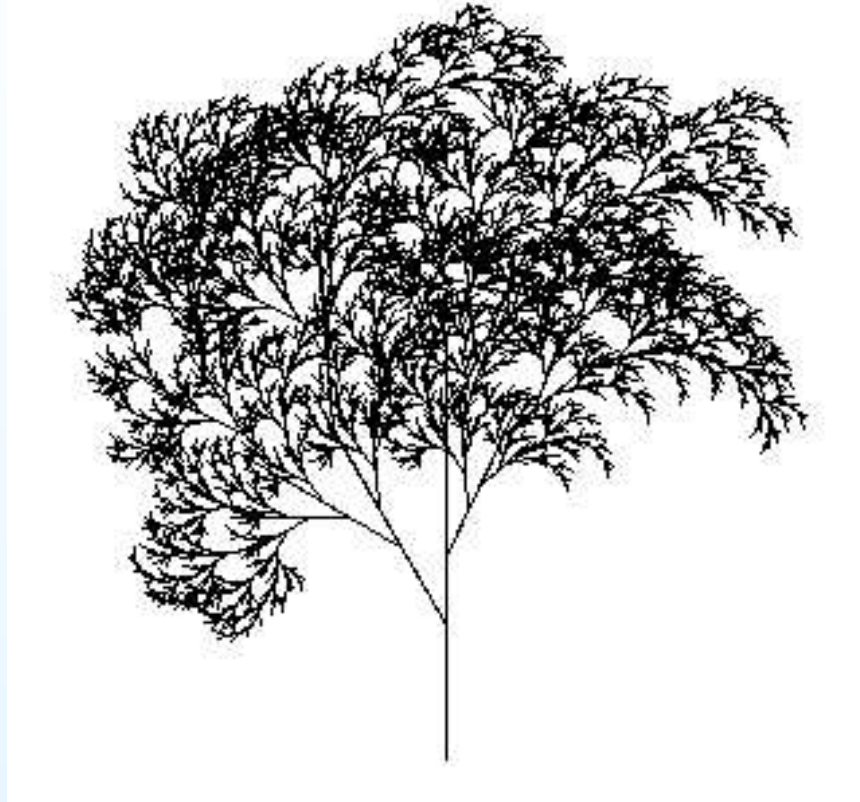
Oorzaak balans?

- lengte verkleinen op zelfde manier → stopvoorwaarde na zelfde aantal stappen

Lengte van takken willekeurig aanpassen

- random getal
 - genereren van een willekeurig geheel getal
 - kleiner dan getal
- bijvoorbeeld
 - `:SIZE*(((random 5) + 5))/10)`

Binaire Bomen in Logo - Natuurlijke Boom



Hoe?