

ALGORITHMS & DATA STRUCTURES

LES 4 : QUICKSORT EN BOMEN

OPDRACHTEN

THEORIE

Actie	Hoofdstuk	2010	2012
Bestudeer	Sorting Algorithms	8.6	8.6
Bestudeer	Trees	18	17

THEORIEOPGAVEN

OPGAVE 1

Sort the sequence 4, 9, 5, 1, 2, 8, 3, 6, 7 by using quicksort with median-of-three pivot selection and a cutoff of 2.

OPGAVE 2 (8.7A)

When the input has been sorted in reverse order, what is the running time of Quicksort?

OPGAVE 3 (8.14A)

When all keys are equal, what is the running time of Quicksort?

OPGAVE 4 (8.15A)

When the input has been sorted, what is the running time of Quicksort?

OPGAVE 5

Teken van de volgende expressies de bijbehorende binaire boom en geef de prefix- en de postfix-vorm van de expressie.

a) $6 * (8 - 3)$

b) $\frac{7-5}{4+6}$

c) $a * (b - c) + \frac{6}{d}$

PRAKTIJK

OPDRACHT 1

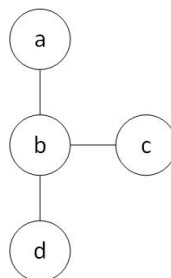
Implementeer het QuickSort-algoritme met behulp van de code uit het boek. Gebruik hiervoor de gegeven interface en unittests. Vergelijk de tijden met de sorteeralgoritmen van vorige week.

OPDRACHT 2 (KAN NODIG ZIJN VOOR PRACTICUM!)

- a) Implementeer de klasse `FCNSNode<T>` die een node representeert van een First Child - Next Sibling boom. Gebruik hiervoor het bijgeleverde interface. Implementeer de volgende twee constructors:

```
public FCNSNode(T data)
public FCNSNode(T data,
                FCNSNode<T> firstChild,
                FCNSNode<T> nextSibling)
```

- b) Implementeer de klasse `FirstChildNextSibling<T>` volgens bijgeleverde interface. De method `Size()` retourneert het aantal nodes van de boom. De method `PrintPreOrder()` print de inhoud van de boom (gekanteld) naar de console. Gegeven de volgende boom:



Deze boom is geïmplementeerd in `DSBuilder.cs`. De uitvoer van `PrintPreOrder()` van deze boom zou als volgt moeten zijn:

```
a
  b
    d
  c
```

- c) Implementeer een `ToString()`. Deze method retourneert `NIL` in het geval van een lege boom. In de andere gevallen wordt het volgende geretourneerd:

```
<data>,FC(<contents first child>),NS(<contents next sibling>)
```

Voor de gegeven voorbeeldboom wordt de uitvoer:

```
a, FC(b, FC(d), NS(c))
```

- d) Implementeer in `DSBuilder.cs` de method `CreateFirstChildNextSibling_18_3()`, een implementatie van de boom van afbeelding 18.3 uit het boek.
- e) Test je datastructuur met de bijgeleverde unittests.

OPDRACHT 3 (KAN NODIG ZIJN VOOR PRACTICUM!)

- a) Implementeer de klasse `BinaryTree<T>`, die een binaire boom representeert. Gebruik hiervoor het bijgeleverde interface. Implementeer de volgende twee constructors:

```
public BinaryTree()  
public BinaryTree(T rootItem)
```

Raadpleeg het boek voor de implementatie van `Merge()`.

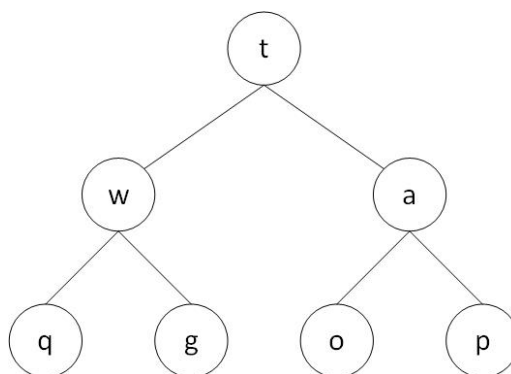
- b) Er is geen standaard `ToString()`, maar drie alternatieve methods dienen geïmplementeerd te worden. De methods geven een stringrepresentatie weer van een traversal volgens de prefix-, infix- of postfix-methode:

```
string ToPrefixString();  
string ToInfixString();  
string ToPostfixString();
```

De uitvoer van deze methods is als volgt:

- In geval van een lege boom: "NIL"
- In de andere gevallen: de string "[", gevolgd door de juiste traversal (gescheiden door een spatie), gevolgd door "]".

Gegeven de volgende boom:



Deze boom is al geïmplementeerd in `DSBuilder.cs`. Het aanroepen van de drie methods op deze boom, zou het volgende resultaat moeten opleveren:

```
[ t [ w [ q NIL NIL ] [ g NIL NIL ] ] [ a [ o NIL NIL ] [ p NIL NIL ] ] ]  
[ [ [ NIL q NIL ] w [ NIL g NIL ] ] t [ [ NIL o NIL ] a [ NIL p NIL ] ] ]  
[ [ [ NIL NIL q ] [ NIL NIL g ] w ] [ [ NIL NIL o ] [ NIL NIL p ] a ] t ]
```

- c) Implementeer in `DSBuilder.cs` de method `CreateBinaryTreeInt()`. Zie de code in `DSBuilder.cs` voor de opbouw van deze boom.

OPDRACHT 4 (18.11)

Implementeer de volgende methods van het interface `IBinaryTree<T>` in `BinaryTree<T>`:

- a) `int NumberOfLeaves()` : het aantal leaves in een boom
- b) `int NumberOfNodesWithOneChild()` : het aantal nodes in een boom dat precies 1 kind heeft
- c) `int NumberOfNodesWithTwoChildren()` : het aantal nodes in een boom dat precies 2 kinderen heeft

Zie ook het interface `IBinaryTree.cs`.