

# EE 2331 Data Structures and Algorithms, Semester B, 2009/10

## Tutorial 9: Binary Trees

Week 9 (18<sup>th</sup> March, 2010)

*The tasks of tutorial exercises are divided into three levels. Level 1 is the basic tasks. You should have enough knowledge to complete them after attending the lecture. Level 2 is the advanced tasks. You should be able to tackle them after revision. Level 3 is the challenge tasks which may be out of the syllabus and is optional to answer. I expect you to complete task A in the tutorial.*

### Outcomes of this tutorial

1. Able to understand the implicit array representation of binary trees
2. Able to implement basic tree operations using recursion

Binary trees are a commonly used data structure. You have learnt in the lecture that for a binary tree,  $n_0$  (the number of nodes with degree 0) is equal to  $n_2$  (the number of nodes with degree 2) plus one. In this tutorial, you are asked to write a program to verify this property.

The implicit array tree is stored in a text file with the following format. We will use the dot character '.' to denote the "absent" nodes of the implicit array representation of a binary tree.

Row	Content	Remark
1 <sup>st</sup>	$C_1 C_2 C_3 \dots C_n$	The implicit array representation of a binary tree $C_i$ is the $i^{\text{th}}$ element (char type) of the binary tree

The structure of tree is defined as follows:

```
#define MAX_TREE_SIZE 64 //The max. size of the array
typedef char TreeElement; //define the tree element (char)
typedef struct _tree { //define the tree structure
    TreeElement data[MAX_TREE_SIZE]; //store the data of the tree
} Tree;
```

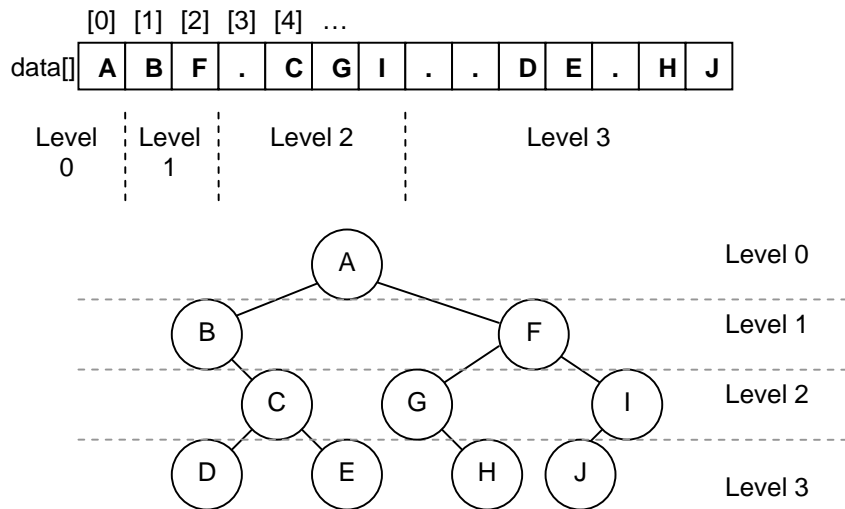


Figure 1. An example of implicit array tree representation.

#### Task A (Level 2): Count the number of nodes with degree 2

Design the recursive function **int countNodeWithDegree2(TreeElement data[], int index)** to count and return the number of nodes with degree 2 in the given binary tree. “data[]” is the implicit array representation of the tree. “index” is the index of the current node in the recursive function.

#### Task B (Level 2): Count the number of nodes with degree 1

Design the recursive function **int countNodeWithDegree1(TreeElement data[], int index)** to count and return the number of nodes with degree 1 in the given binary tree. “data[]” is the implicit array representation of the tree. “index” is the index of the current node in the recursive function.

#### Task C (Level 2): Count the number of nodes with degree 0

Design the recursive function **int countNodeWithDegree0(TreeElement data[], int index)** to count and return the number of nodes with degree 0 in the given binary tree. “data[]” is the implicit array representation of the tree. “index” is the index of the current node in the recursive function.

Expected Output:

Enter the file name for testing: test1.txt  
The array tree is: [ABF.CGI..DE.HJ]

The number of nodes with degree 2 = 3  
The number of nodes with degree 1 = 3  
The number of nodes with degree 0 = 4

" $n_0 = n_2 + 1$ " is correct!

Enter the file name for testing: test2.txt  
The array tree is: [ABF.CGI..DE..J]

The number of nodes with degree 2 = 3  
The number of nodes with degree 1 = 2  
The number of nodes with degree 0 = 4

" $n_0 = n_2 + 1$ " is correct!

Enter the file name for testing: test3.txt  
The array tree is: [EFGXYCT..QZSWVU]

The number of nodes with degree 2 = 6  
The number of nodes with degree 1 = 0  
The number of nodes with degree 0 = 7

" $n_0 = n_2 + 1$ " is correct!

Enter the file name for testing: test4.txt  
The array tree is: [ABF.CGI..XDE..J]

The number of nodes with degree 2 = 3  
The number of nodes with degree 1 = 3  
The number of nodes with degree 0 = 4

" $n_0 = n_2 + 1$ " is correct!

Enter the file name for testing: test5.txt  
The array tree is: [EFGXYCT..QZ.VWR]

The number of nodes with degree 2 = 5  
The number of nodes with degree 1 = 1  
The number of nodes with degree 0 = 6

" $n_0 = n_2 + 1$ " is correct!

**Discussion:** These functions are now implemented by recursion. Can they be implemented by iteration? Any extra data structures do you need?