UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 212 Tutorial 3: Version B

- 02/03/2012
- 50minutes
- 3 questions for a total of 33 marks.

Name:_____

Student/staff Nr:_____

Marker (office use):_____

Question 1 Trees ..... (6 marks)

1.1 How is the level of a node calculated? (1)

> **Solution:** num arcs in path + 1

Answer:

1.2 What is the maximum number of nodes that can be contained in a Binary Search Tree of height 7? (1)

> **Solution:** $2^7 - 1 = 127$

Answer:

1.3 Assume you were given the following Binary Tree Node class: (4)

```
public class BinaryNode<T extends Comparable<T>>
{
        public BinaryNode(T element)
        {
                key = element;
                left = right = null;
        }

        protected T key;
        protected BinaryNode<T> left,right;
}
```

Write a recursive method `countInternals` which will count the number of non-leaf nodes in a Binary Tree. The method should receive only parameters that are absolutely necessary.

Solution:

```
//Suggestion (non-shorthand)...
public int countLeafs(BinaryNode<T> n)
{
        if(n==null)
                return 0;
        if(n.left == null && n.right == null)
                return 0;
        if(n.left != null || n.right != null)
                return 1 + countLeafs(n.left) + countLeafs(n.right);
}

/*
1 for correct signature
3 marks, one for each return and it's correct condition for
returning as such...
```

Answer:

Question 2 Binary Search Trees ..... (15 marks)

Consider the Binary Search Tree of figure 1 and answer the following questions:

2.1 Give the order in which nodes would be visited if post-order depth first search were applied: (7)

> **Solution:** 1/2 mark for each node in its correct position in the sequence...
>
> F,D,I,L,J,H,N,O,P,U,Z,W,S,M

Answer:

2.2 In Binary Search Trees, the direct **successor** of a key/element k_i is the element k_j that will be processed directly after k_i. In other words, k_j will directly follow k_i if an in-order processing is applied to the tree (for example when printing out keys). Write a Java method `BinaryNode<T> findSucc(BinaryNode<T> pred)` which will find and return the node containing the key which is the direct successor to the key held by the node passed in as a parameter. In all circumstances that no such node is found, `null` should be returned. (3)

> **Solution:**
>
> ```
> //Suggestion
> BinaryNode<T> findPred(BinaryNode<T> pred){
>         if(pred == null)
>                 return null;
>         BinaryNode<T> n = pred.right;
>         while(n.left!=null)
>                 n = n.left;
>         return n;
> }
>
> 1 mark for returning null under all required conditions
> 1 for stepping one left in the tree
> 1 for going as far right as possible from that point on
> ```

Answer:

2.3 The direct **predecessor** of a key k_i is the largest key k_j in a tree that is still smaller than k_i. Apply delete by merging to the tree in figure 1 and remove node **S**. You must make use of the direct **predecessor** where applicable and for your answer only draw the final tree. (3)

Solution: 1 mark P is M's right child
1 mark W is P's right child
1 mark, apart from these changes the rest of the tree remains the same

Answer:

2.4 Using any of the depth first traversals for binary trees as discussed in the textbook, what would be the result if `visit` were only applied to leaf nodes? Motivate your answer. (2)

> **Solution:** The order of the nodes will be the same for all traversals. Comes down to when a node is visited in relation to its children, this changes the sequence of the keys output. Left always happens before right, leaf's parents won't be visited to insert additional keys into the sequence, leafs don't have children to insert keys into the sequence...

Answer:

Question 3 Threaded Trees ..... (12 marks)

3.1 Assume you are implementing a `threaded tree` to be used for `post-order` traversals. The right reference of each node either refers to a right child OR references the successor of the node in the traversal, the meaning of which is indicated by a `hasSuccessor` flag for each node. The right reference may also be null if there is no right child or successor. Successor links should be indicated with a dashed line as in the textbook example. Insert the following keys, in the given order, into such a tree and redraw the tree after every insert: (7)

`51,60,70,40,20,45,15`

Answer:

3.2 Write the pseudo code to traverse a threaded tree where the threads were established such that a post-order traversal is possible. (5)
Answer:

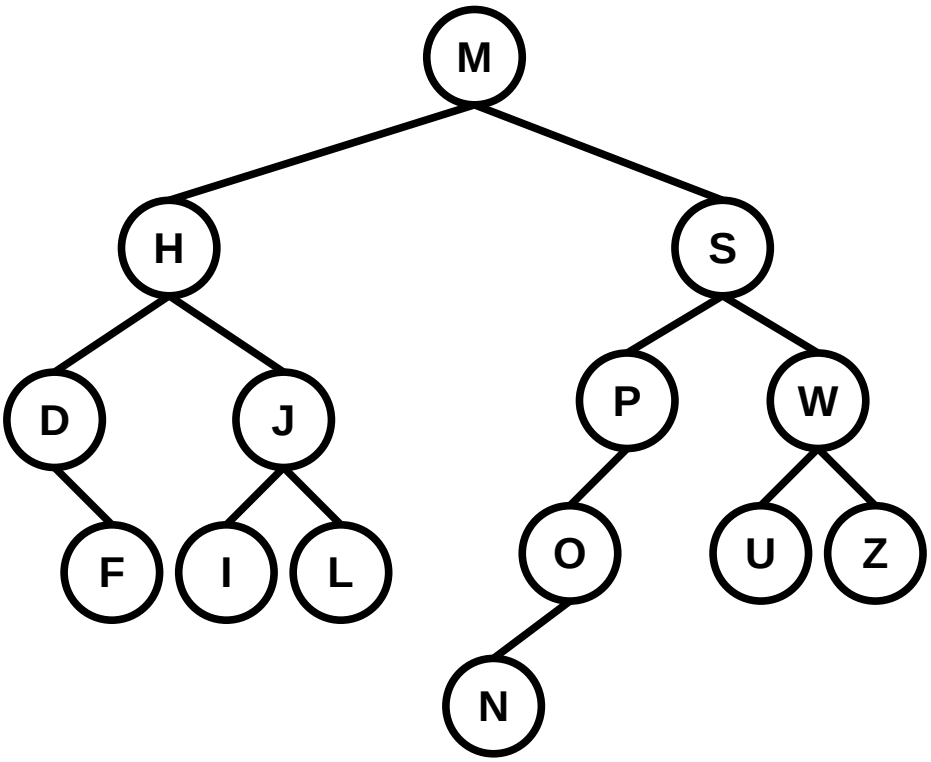

Figure 1: Binary Tree