**Review Problems on Trees: Chapter 14: pp 561 – 615.**

* Be able to apply tree terminology (parent, child, root, leaf, siblings, ancestors, descendants, subtree , height, binary tree, left (right) child of node nm left (right) subtree of node n, binary search tree, empty binary tree, full binary tree, complete binary tree)balanced binary tree.
* Be able to traverse a binary tree in preorder, inorder and postorder.
* Be able to store a binary tree using an Array (computational strategy) or as a referenced based representation
* Understand how the operations (insert, delete, retried, traverse and search) are implemented and **the best and worst behavior**.

**Text: Self-Text Exercises pp 632- 633: 1-11.**  The Answers are given in the back of the book.

!. Arrange nodes that contain the letters D, Q, X, F, S, G and W into 2 binary search trees: one that has maximum height and one that has minimum height.

|  |  |
| --- | --- |
| **A binary search tree with maximum height** (7 nodes and height of tree = 7)  D  F  G  S  Q  W  X | **A binary search tree with minimum height**: a full tree that has height 2.  S    F W  D G Q X |

2. Draw the Binary search tree that would result in the addition of adding the integers in the following order: 10, 15, 5, 18, 14, 6, 20 & 9.

10

15

5

6

14

18

9

20

a) Using this BST, trace the method of searching the BST for the search key of 12, by listing the nodes in the order in which a search would visit them.

Visit 10; Visit 15 Visit 14 [Since left subtree is null, algorithm ends in failure]

b) Using this BST,

* show the BST that results when you delete the node that contains 10.

14

15

5

6

18

9

20

ii.) Describe the steps in the method that would need to be used to effect the above

* Overall since the node 10 (which we removed) had two children, we would need to the node containing 10 with they the node in the right subtree with the smallest value.
* 1) Find the node that we want to replace.
* 2) Find the node that contains the smallest value: Go to right Subtree. Find the leftmost child of this subtree.
* 3) copy the value of leftmost child into the node that contained 10
* 4) delete the leftmost child.

3) Evaluate the following prefix notation expression. Show all steps:

\* / - 10 2 4 - + 3 5 1

\* / - 10 2 4 - + 3 5 1

= \* / 8 4 - + 3 5 1

= \* 2 - + 3 5 1

= \* 2 – 8 1

= \* 2 7 = 14

4) Evaluate the following postfix notation expression: Show all steps:

4 2 + 3 5 1 - \* +

4 2 + 3 5 1 - \* +

6 3 5 1 - \* +

6 3 4 \* +

6 12 + = 18

5) Binary Trees: Given the following expression tree

|  |  |
| --- | --- |
|  | a) Find the postfix notation:  2 3 \* 2 1 - / 5 4 1 - \* +  b) Find the prefix notation:  + / \* 2 3 – 2 1 \* 5 – 4 1  c) Evaluate one of the above.  21 |

6) Given the following tree

|  |  |
| --- | --- |
|  | a) Is this a binary tree? yes  b) Is this a binary search tree? \_\_no\_\_\_  c) Is this tree full? \_\_\_no\_\_\_\_\_  [Note: there are only 2 leaves at level 4 and 3 leaves at level 3. In a full tree, all leaves are at the same level.]  d) Is this tree complete? \_yes\_\_\_\_\_  e) What is the height of this tree? \_\_4\_  f) is this tree balanced? \_yes\_\_\_\_\_\_ |

f) Represent this tree as an array: [Use the computational strategy]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| + | \* | / | - | c | d | e | a | b |  |

7. List an advantage and disadvantage of using a referenced based implementation of a Binary Tree over using an Array (computational strategy) implementation of a Binary Tree.

|  |  |  |
| --- | --- | --- |
|  | Advantage | Disadvantage |
| Referenced Based Implementation | Can be used with any trees. | Each node requires 2 additional fields for the left and right references) that take up extra space |
| Array (computational strategy) | Easy to find the left or right child of a parent by knowing the index of the parent. If index of parent is I, left child is a index 2i + 1, while right child is at index 2i + 2.  Given a child, easy to find index of the parent. If I is child, then  Parent is theindex which is the integer value of ( I-1)/2)  The Array | Only useful if tree is complete tree.  An Array could possibly run out of space |

8. Describe the best and worst behavior for inserting an item into a BST.

The best behavior is when the **BST is balanced** (i.e. when the left and right subtrees of any node have heights that differ by at most 1. Note: This always happens with a full tree (subtrees differ in height by at most 1) and with a complete tree (subtrees of any node have the same height.)) When the BST is balance, inserting an item is O(logn)).

**The worst behavior is when the BST consists of a tree that only has right children, or only have rleft children**. The height of this tree of n nodes is n, but the right (or left) subtree has height n-1, while the left (or right) subtree has height 0.

**In this case, inserting an item is O(n).**