

Problem No. 1

In what order a complete binary tree of height 4 (15 nodes) will be traversed, by invoking 'A'. Use the first 15 natural numbers to create the tree. Assume visit(node) only print the node.

<pre>function A (TreeNode<TYPE> *node) { if (node) { B (node → right); B (node → left); visit (node); } }</pre>	<pre>function B (TreeNode<TYPE> *node) { if (node) { A (node → left); A (node → right); visit (node); } }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Problem No. 2

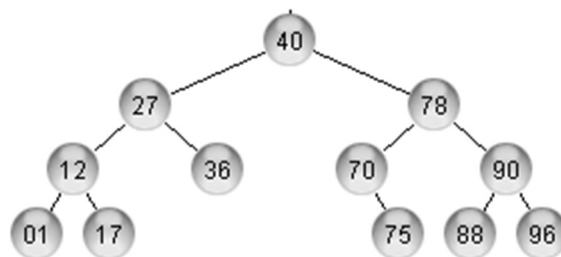
Assume A is an "Array-Based Tree" with 50 members. Answer the following questions to the point:

- a) Is A[23] a leaf node?
- b) What is the index of the first leaf node?
- c) Which element of A is the parent of A[32]?
- d) What is the height of the tree?
- e) Does any item have exactly one child?

Problem No. 3

Answer the following question with all necessary work provide as part of the solution.

Consider the following BST:



- a) Draw the resulting tree if we insert 28 and 41 in the given tree.
- b) Draw the resulting tree if we now delete 78 and 40 respectively.
- c) Draw the resulting tree if the root node of the tree in part (b) is deleted.
- d) List all nodes which will be traversed if we need to find 45.
- e) Compute and illustrate the balance factor of each node of the tree.

Problem No. 4

Write a function that takes a pointer to a BST and count all nodes that have any of the child (all parent's nodes). The signature of the function given below:

```
unsigned int countParentNodesInBST(const BTreeNode<T> *Tree)
```

Problem No. 5

Write a function that takes pointers to 2-distinct BST and return true if the two trees are identical in structures and values of the nodes.

```
unsigned int IsIdenticalwithValues(const BTreeNode<T> *Tree,  
const BTreeNode<T> * T2)
```

Problem No. 6

Write a function that takes two Binary Trees (BT) pointers and check if the two BT are structural identical or not.

```
bool IsIdenticals(const BTreeNode<T> *T1, const BTreeNode<T> * T2)
```

Problem No. 7

Given the following signature of the function. Write the code to give the next Lower value than the given x for the tree given by the pointer root.

```
T& BSTree<T>::NextLower(BSTree<T> * root, T & x)
```

Problem No. 8

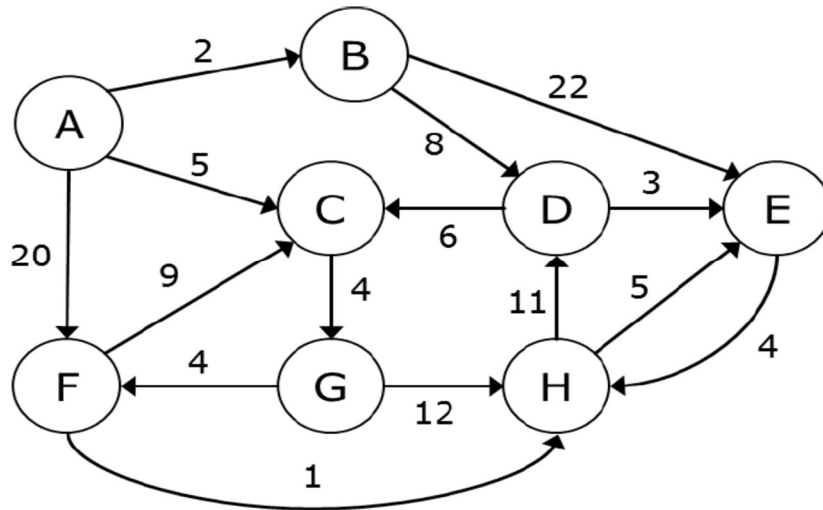
Insert the following numbers into an initially empty BSTree and an initially empty AVLTree.
21, 43, 11, 5, 98, 65, 33, 18, 7, 3

Problem No. 9

Insert the following numbers into an initially empty Multiway Tree take B-Tree of order of order 2-3, show all steps of the insertion and maintenance.
21, 43, 11, 5, 98, 65, 33, 18, 7, 3

Problem No. 10

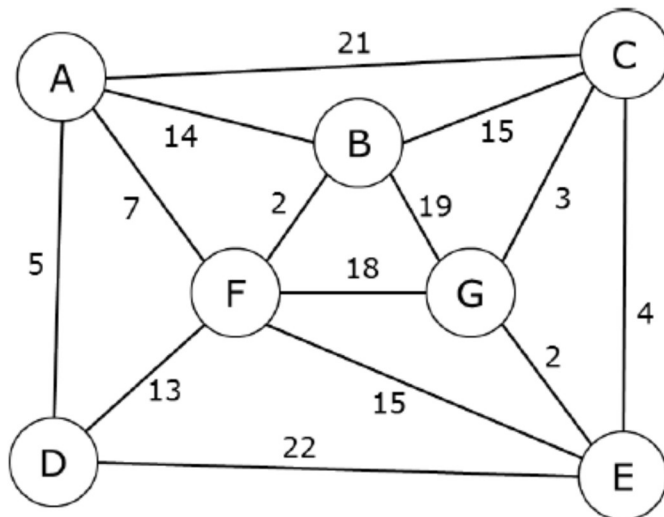
Consider the following given graph.



Use Dijkstra's algorithm to calculate the single-source shortest paths from vertex A to every other vertex.

Problem No. 11

Consider the following given graph.



Find the minimum spanning tree with Prim's and Kruskal's algorithm.