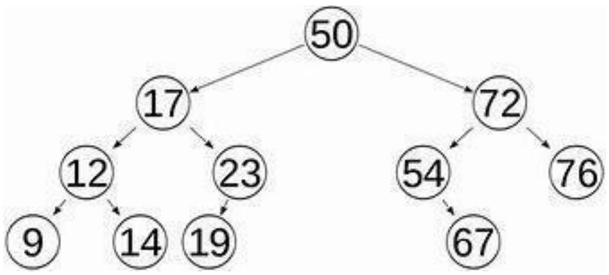
6# Look at the diagram of the BST below:

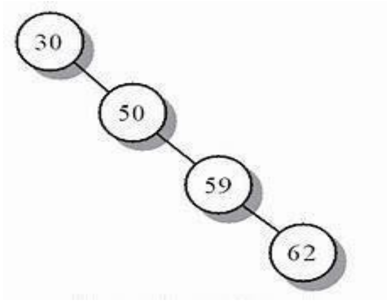


Which of the following input sequences would generate the above BST assuming the add() algorithm does NOT do any balancing, but just inserts the nodes into the tree as given. PICK ONE

A	50 17 72 12 23 54 76 9 14 19 67
B	50 17 12 9 14 23 19 72 54 67 76
C	50 17 14 23 9 12 19 72 67 54 76
D	50 17 14 23 19 9 12 76 72 67 54
E	A & B
F	A & C
G	A & D

7#	T	The above BST (#6) is height balanced
8#	F	The above BST (#6) is full
9#	F	The above BST (#6) is complete
10#	T	The complexity of .add() .remove() and .contains() on the above tree (#6) is much closer to log2N

11# Look at the diagram of the BST below:



Which of the following sequences would generate the above BST? PICK ONE

A	52 59 30 62
B	62 59 50 30

C 30 50 59 62

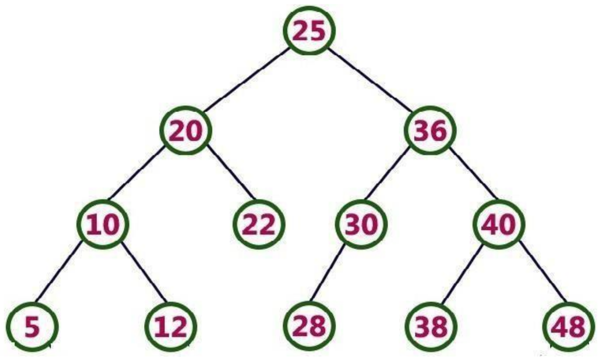
12# F The above BST (#11) is height balanced

13# F The above BST #11) is full

14# F The above BST (#11) is complete

15# F The complexity of .add() .remove() and .contains() tree (prob #11) is much closer to log2N than N

Look at the diagram of the BST below:



THE NEXT 3 QUESTIONS APPLY TO THE BST ABOVE

16# List the nodes PRE ORDER separated by space on single line on the #16 line of your answer file

25 20 10 5 12 22 36 30 28 40 38 48

17# List the nodes POST ORDER separated by space on single line on the #17 line of your answer file

5 12 10 22 20 28 30 38 48 40 36 25

18# List the nodes LEVEL ORDER separated by space on single line on the #18 line of your answer file

25 20 36 10 22 30 40 5 12 28 38 48

19# Suppose you want rebalance a very lopsided BST by copying the elements into a second tree which will become the balanced version of the bad tree. You copy all the elements into an array, sorted the array, then visited the middle of the array, then the middle of the left half, then the middle of the right half recursively into the middles of those halves a binary search like visitation pattern, inserting each element of that array into the 2nd tree.

Let's assume you already have the sorted array to start with – and only counting the cost AFTER you are given the sorted array: what is the complexity of traversing that array in a binary search like pattern and doing an add into the other tree as the visitation operation on each element of the array? PICK ONE

A log2N

B N * log2N visiting all n elements from array once * cost of insertion into balanced tree

C N squared

D n squared * log2N