**BINARY SEARCH TREE**

1)

a)

pre-order : D,B,A,C,F,E,G

in-order : A,B,C,D,E,F,G

post-order : A,C,B,E,G,F,D

level-order : D,B,F,A,C,E,G

b)

pre-order : C,B,A,D,E

in-order : A,B,C,D,E

post-order : A,B,E,D,C

level-order : C,B,D,A,E

c)

pre-order : E,C,B,A,D,H,F,G,I

in-order : A,B,C,D,E,F,G,H,I

post-order : A,B,D,C,G,F,I,H,E

level-order : E,C,H,B,D,A,F,I,G

2)   
**For sequence (1): 4, 3, 1, 11, 5, 9, 2, 6, 15, 12**

Preorder traversal: 4, 3, 1, 2, 11, 5, 9, 6, 15, 12

Inorder traversal: 1, 2, 3, 4, 5, 6, 9, 11, 12, 15

Postorder traversal: 2, 1, 3, 6, 9, 5, 12, 15, 11, 4

Level-order traversal: 4, 3, 11, 1, 15, 2, 5, 12, 9, 6

**Deleting keys 2, 3, and 11**:

**After deleting key 2**:

Preorder traversal: 4, 3, 1, 11, 5, 9, 6, 15, 12

Inorder traversal: 1, 3, 4, 5, 6, 9, 11, 12, 15

Postorder traversal: 1, 3, 6, 9, 5, 12, 15, 11, 4

Level-order traversal: 4, 3, 11, 1, 15, 5, 12, 9, 6

**After deleting key 3** :

Preorder traversal: 4, 11, 1, 15, 12, 5, 9, 6

Inorder traversal: 1, 4, 5, 6, 9, 11, 12, 15

Postorder traversal: 1, 6, 9, 5, 12, 15, 11, 4

Level-order traversal: 4, 11, 1, 15, 5, 12, 9, 6

**After deleting key 11** :

Preorder traversal: 4, 12, 1, 15, 5, 9, 6

Inorder traversal: 1, 4, 5, 6, 9, 12, 15

Postorder traversal: 1, 6, 9, 5, 15, 12, 4

Level-order traversal: 4, 12, 1, 15, 5, 9, 6

**For sequence (2): 12, 7, 1, 3, 2, 5, 10, 8, 6, 9**

Preorder traversal: 12, 7, 3, 2, 1, 10, 8, 5, 6, 9

Inorder traversal: 2, 3, 7, 1, 8, 10, 12, 6, 5, 9

Postorder traversal: 2, 3, 1, 8, 6, 9, 5, 10, 7, 12

Level-order traversal: 12, 7, 1, 3, 10, 5, 2, 8, 6, 9

Deleting keys 5, 6, and 7:

**After deleting key 5:**

Preorder traversal: 12, 7, 3, 2, 1, 10, 8, 6, 9

Inorder traversal: 2, 3, 6, 7, 8, 9, 10, 1, 12

Postorder traversal: 2, 6, 9, 8, 10, 1, 7, 12

Level-order traversal: 12, 7, 1, 3, 10, 6, 2, 8, 9

**After deleting key 6:**

Preorder traversal: 12, 7, 3, 2, 1, 10, 8, 9

Inorder traversal: 2, 3, 7, 9, 8, 10, 1, 12

Postorder traversal: 2, 3, 9, 8, 10, 1, 7, 12

Level-order traversal: 12, 7, 1, 3, 10, 8, 2, 9

**After deleting key 7:**

Preorder traversal: 12, 8, 3, 2, 1, 10, 9

Inorder traversal: 2, 3, 8, 10, 9, 1, 12

Postorder traversal: 2, 3, 9, 10, 1, 8, 12

Level-order traversal: 12, 8, 1, 3, 10, 9, 2

3)

Key sequence 1: 12, 7, 1, 3, 2, 5, 10, 8, 6, 9

a)

**Inserting keys 11 to AVL tree**

Preorder traversal: 7, 3, 1, 2, 5, 4, 12, 10, 8, 9, 11

Inorder traversal: 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12

Postorder traversal: 2, 1, 4, 5, 3, 9, 8, 11, 10, 12, 7

Level-order traversal: 7, 3, 12, 1, 5, 10, 2, 4, 8, 11, 9

**Key sequence 2: 4, 3, 1, 11, 5, 9, 2, 6, 15, 12**

Preorder traversal: 5, 3, 2, 1, 4, 11, 9, 6, 12, 15

Inorder traversal: 1, 2, 3, 4, 5, 6, 9, 11, 12, 15

Postorder traversal: 1, 2, 4, 3, 6, 9, 15, 12, 11, 5

Level-order traversal: 5, 3, 11, 2, 4, 9, 12, 1, 6, 15

6)

No, the given sequence of nodes: 2, 90, 63, 70, 68, 72, 57, cannot be the sequence of nodes examined when searching for the number 57 in a binary search tree that contains numbers between 1 and 100.

In a binary search tree, the order of examining nodes during a search follows a specific pattern based on the values being searched for. For any given node, the left subtree contains smaller values, and the right subtree contains larger values.

If we consider a binary search tree with numbers between 1 and 100, the search for the number 57 should follow a pattern where nodes with values smaller than 57 are examined first, followed by nodes with values larger than 57.

In the given sequence, we observe that the number 57 is examined before the numbers 63, 70, 68, and 72. This violates the expected order of examination in a binary search tree and suggests that the given sequence does not correspond to the search for the number 57 in the given binary search tree.

Therefore, the given sequence is not a valid sequence of nodes examined when searching for the number 57 in the binary search tree with numbers between 1 and 100.