Preorder Traversal:
10,4,1,2,3,6,5,9,7,15,13,15

Inorder Traversal:
1,2,3,4,5,6,7,9,10,13,15,15

Postorder Traversal:
3,2,1,5,7,9,6,4,13,15,15,10
2.

Comparing 50000 Insert operations...

BST (increasing order) = 24398 ms
BST (random order) = 40 ms
AVL Tree (increasing order) = 17 ms
AVL Tree (random order) = 24 ms
Splay Tree (increasing order) = 27 ms
Splay Tree (random order) = 26 ms

When nodes are inserted in BST tree in increasing order, it shows worst performance (longest running time) due to skew. i.e. O(N). It simply forms a right skewed tree like a list. Ex. 1-2-3-4-5...

For other trees, run-time performance is log(N). AVL and Splay tree balances each time during addition of a node.
3.

**Comparing 10000 Search operations...**

BST (increasing order) = 586 ms  
BST (random order) = 6 ms  
AVL Tree (increasing order) = 2 ms  
AVL Tree (random order) = 4 ms  
Splay Tree (increasing order) = 2 ms  
Splay Tree (random order) = 6 ms

**Comparing 100 x100 Search operations...**

BST (increasing order) = 5 ms  
BST (random order) = 53 ms  
AVL Tree (increasing order) = 1 ms  
AVL Tree (random order) = 1 ms  
Splay Tree (increasing order) = 1 ms  
Splay Tree (random order) = 1 ms

The AVL tree takes $O(\log N)$ for each search and splay tree takes average $O(\log N)$ run time. They are comparable. Splay tree shows shorter run-time when each node is used repeatedly due to splaying effect. The accessed node is brought to the root so it makes it faster for another access $O(1)$. Out of 100*100 search, 100*99 search is performed at $O(1)$ run-time.

**Comparing 100 x10000 Search operations...**

AVL Tree (increasing order) = 103 ms  
AVL Tree (random order) = 119 ms  
Splay Tree (increasing order) = 26 ms
Splay Tree (random order) = 48 ms