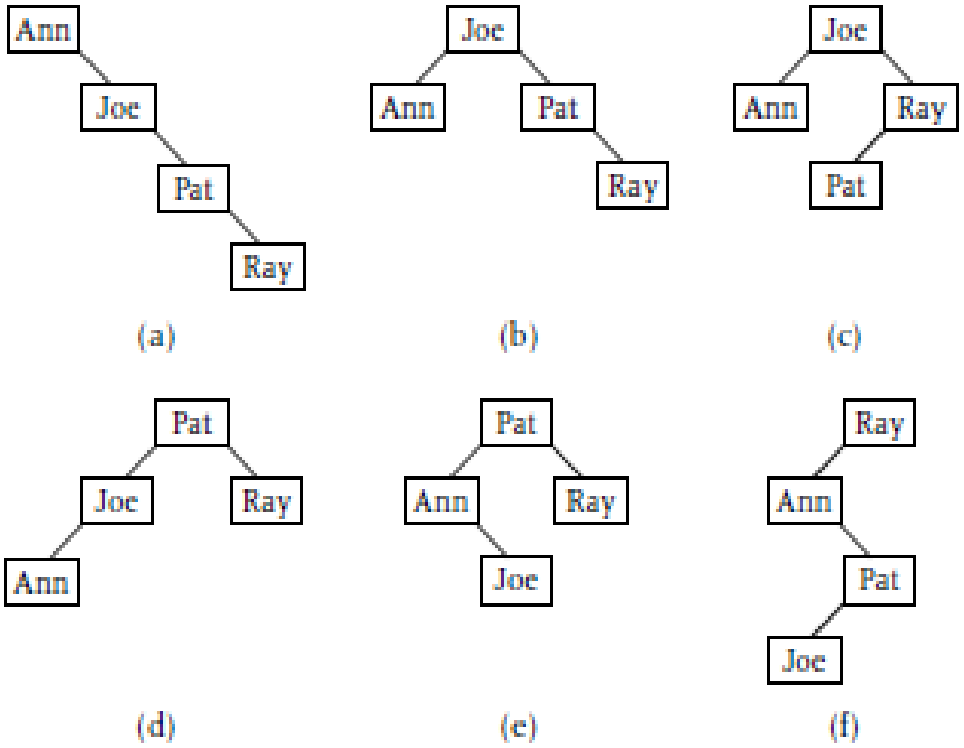
Binary Search Trees (BST)

Textbook Reading:

Chapter 4, Section 4.5, pp. 163-166.

Binary Search Trees (BFT)
Examples for keys Ann < Joe < Pat < Ray



Inorder Traversal

Performing and inorder traversal of a BST will output the keys in sorted order.

Conversely, given any binary tree and assigning a set of keys $K_0 < K_1 < \ldots < K_{n-1}$ to the nodes of a tree in inorder, i.e., scanning the keys and assigning them to the binary search tree using an in order traversal, results in a BST.

PSN. Searching a BST

Write a recursive function SearchBST in C++ for search a binary tree for a Search Key. If found it returns pointer to node where found otherwise it returns NULL.

Assume the nodes of the tree are implemented using the structure:

```
typedef int KeyType;

struct Node {
        KeyType Key;
        Node *Left;
        Node *Right ;

}; //END TREENODE

typedef *Node ptrNode;
```

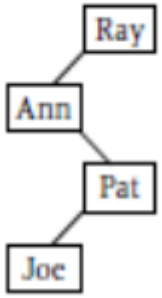
PSN. Inserting into a BST

Write a C++ recursive function InsertBST for inserting into a BST.

Preorder Traversal

Storing records of a BST in sequential file using a preorder traversal allows the BST to be recovered by reading records sequentially from file and inserting into BST, where initial BST is the null tree.

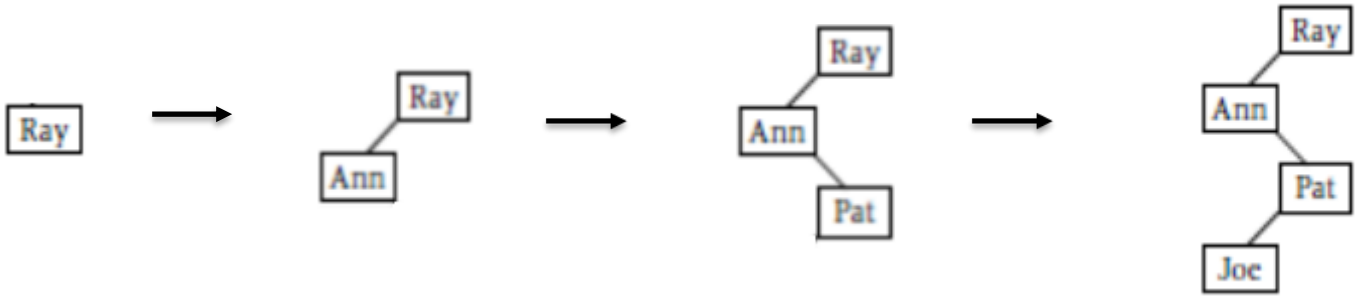
Illustration



Storage in File using Preorder Traversal:

Ray, Ann, Pat, Joe

Reading File Sequentially and inserting



Deleting from a BST

1. If a node to be deleted is a leaf, just delete it.

2. If a node to be deleted has just one child, replace it with that child.

3. If a node to be deleted has two children, replace the **value** of by its in-order successor's value then delete the in-order successor by applying either 1. or 2. (could also use in-order predecessor).

After years of searching, I finally found a great herb joke.

It's about thyme.