COMP 2611 – Data Structures

Lab #3 (September 25-27, 2023)

Recursion with Arrays / Binary Trees

Instructions

Download `Lab3-Files.zip` and unzip the archive. You will obtain two sub-folders, each of which contains a Dev-C++ project. Question 1 requires you to open the `Array.dev` project from the `Array` folder and write several functions in the file, `Array.cpp`. Question 3 requires you to open the `BinaryTree.dev` project from the `BinaryTree` folder and write several functions in the file, `BinaryTree.cpp`.

Recursion with Arrays

1. Folder: Array

The file `Array.h` contains prototypes for the following recursive functions:

Function	**Description**

`void printArrayRec`
`    (int a[], int start, int n);`
	Displays the elements of the array on the monitor.

`bool containsArrayRec`
`    (int a[], int start, int n, int key);`
	Returns *true* if *key* is present in the array and *false*, otherwise.

`int sumArrayRec`
`    (int a[], int start, int n);`
	Returns the sum of the elements in the array.

`int maxArrayRec`
`    (int a[], int start, int n);`
	Returns the maximum element in the array or INT_MIN if the array is empty.

`bool binarySearchRec`
`    (int a[], int start, int end, int key);`
	Assuming that the elements of the array are in ascending order, returns *true* if *key* is present in the array and *false*, otherwise.

In the first four function prototypes above, *n* is the number of elements in the array *a*, and *start* is the index of the array to be used as the starting point in each recursive call.

In *binarySearchRec*, *start* and *end* are the starting and ending locations to perform the binary search.

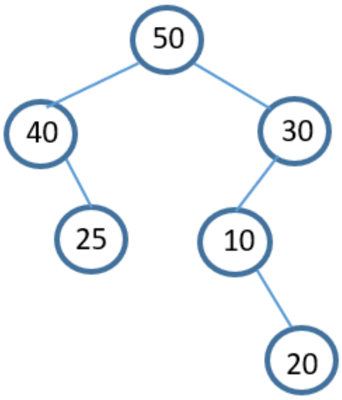
(a) Write the code for each of the functions above in `Array.cpp`.

(b) Code has already been written in `UsingArray.cpp` to test the five functions. Compile and test the program. Ensure that the correct results are obtained.

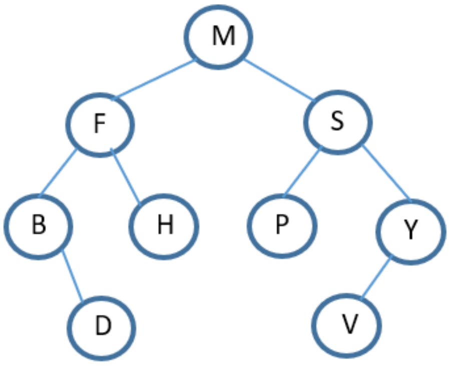
Binary Trees

2. Give the preorder, inorder, and postorder traversals of the following binary trees:

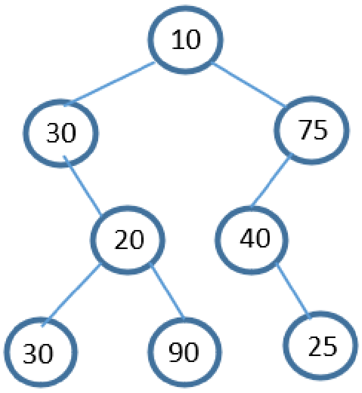
(a)



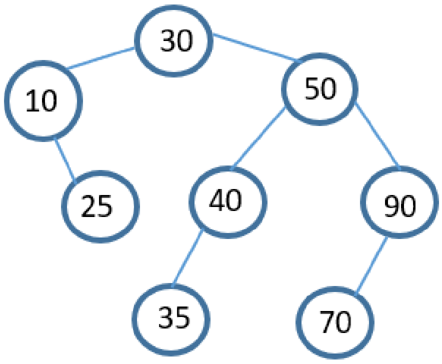
(b)



(c)



(d)



3. (a) In `BinaryTree.cpp`, write the code for the *createBTNode* function with following prototype:

```
BTNode * createBTNode (int n);
```

(b) In `BinaryTree.cpp`, write the code for the *preOrder*, *inOrder*, and *postOrder* functions with the following prototypes:

```
void preOrder (BTNode * root);
void inOrder (BTNode * root);
void postOrder (BTNode * root);
```

The functions must all be recursive and should simply display the value stored in the node when it is “visited”.

(c) In the *main* function of `UsingBinaryTree.cpp`, write code to create the binary tree shown in Question 2(d). Set the value of *root* to Node 30. Code has already been written to create Node 30 and Node 10 and connect them as shown in the diagram.

(d) Call the *preOrder*, *inOrder*, and *postOrder* functions with the value of *root* and ensure that the results obtained correspond with your answer for Question 2(d).

End of Lab #3