Name: *John Hellrung*

This reflection is to be completed individually, although consultations with TAs and classmates are encouraged as long as they are appropriately acknowledged.

This assignment is designed for you to work with binary trees implemented in C++ and uses recursion. This reflection document is intended to help you think about the potential challenges involved from converting a program from one language to another and about recursion and trees in general.

#### Memory Management

Notice that the C++ version of the program contains code that explicitly allocates and deallocates memory. In the box below, give two examples of when space for two different data structures was allocated. With each example, explain exactly what the code is doing and why it is necessary:

|  |
| --- |
| In line 59 of the C++ code there is int\* leftArray = new int[index] this is allowing memory for a temp array that will store new nodes or be made into a leaves. It is needed because without this method the code could not produce new notes or leaves. In line 17 of the C++ there is a line the allow memory in int. (int data;) This line creates memory called a variable which is used for resenting the data of a note in the tree the program is trying to build. This is important because without the program could not store data for a note when called. |

In the box below, give two examples of when space used by the program for two different data structures was deallocated. What is the main possible consequence for not deallocating memory properly?

|  |
| --- |
| The program deallocate memony on 77 and 65 in the function called “populate\_from\_array” and in line 105 in the function called “delete\_node”. In these function they have lines to where they deallocate memory. In the function called “populate\_from\_array” it used deallocation to delate old temp array which are used to help store memory that go into a node of a tree. This is import because without deallocation memory part in the function you are a hard time reallocating and same memory which is used is the function and it’s also bad programing. Next is the function called “delete\_node” it is designed to delete the node from the tree if needed. That being said the function takes the current\_node it is a variable is a repetition of memory and deallocate it from the tree and so run the program to delete children notes. This is important because that needs to be removed because is the main pursue of the function. |

Given all the information above, what kinds of problems should you look out for when converting a program that uses dynamically allocated data structures from Python into C++?

|  |
| --- |
| You should lookout for when deleting memory and adding memory because their completely different is python. They needed to modify to work in the python code. |

#### Recursion and Trees

Consider the code for finding a value in the tree. In the box below, describe how the recursive function calls work to see if the value 8 is in the binary search tree. Be sure to explain the how the runtime stack changes for each recursive call:

|  |
| --- |
| The code calls the functions called populate\_from\_array about 7 times. Each time the function is call it stacks an array for data and each node for the next even until it each 12. |

Describe how the recursive function calls work to find if the value 1 is in the binary search tree. Be sure to explain the how the runtime stack changes for each recursive call:

|  |
| --- |
| First the find function is called and it finds that the targetValue less than the currentNode->data and now it call again for another 5 times until it find that the tree does hold the value 1. |