# horizontal linehorizontal lineData Structures

Homework Assignment 6 - Tree

Problem 1 – AST Dump to Tree - 25 Points

Problem 2 – Linked Binary Tree - 25 Points

Problem 3 – Isomorphic - 25 Points

Problem 4 – One-Zero-Tree - 25 Points

**Notes and Requirements**

* Your submission must be your effort. You can not copy other students' code.
* This worksheet is graded on performance; Implementations must be correct.
* You are encouraged to visit our office hours to ask coding questions.
* Only the latest (most recent) submission is graded.
* Late submissions are not considered for grading.
* You can not use any third-party libraries.

**Some assignments on this worksheet are manually graded.**

## Problem 1 – AST Dump to Tree - 25 Points

Understanding how to analyze and reconstruct abstract syntax trees is essential for building interpreters, static analyzers, and automated grading tools. This exercise will give you practical experience with tree structures and parsing structured data without relying on the original source module.

**You are given the output of Python’s ast.dump(..., indent=3) as a string. Your task is to parse the indented string into a custom TreeWithoutParent structure.** You are not allowed to use the ast module or any other library to reconstruct nodes.

You can use the pretty\_print() function to visualize your tree. For example, this code:

while b != 0:

if a > b:

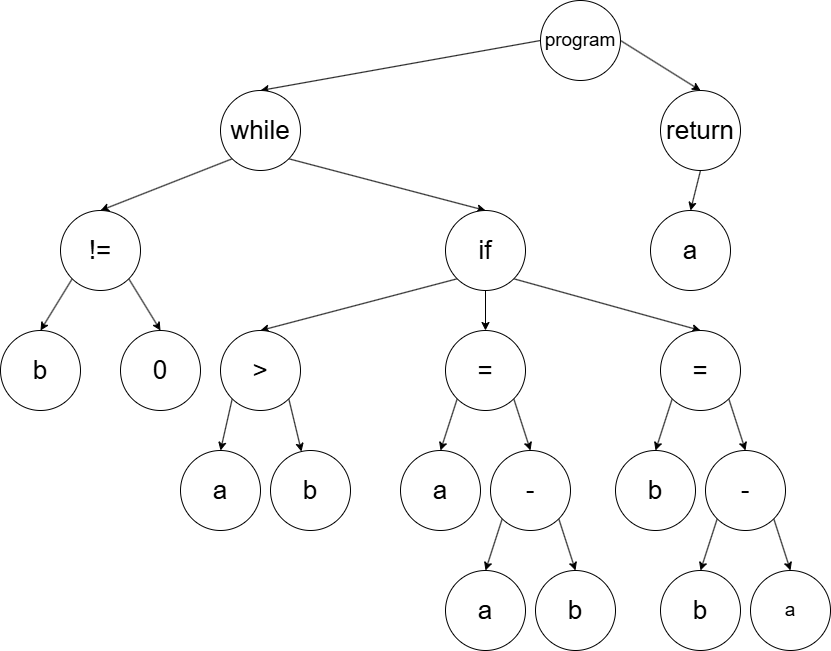
a = a - b

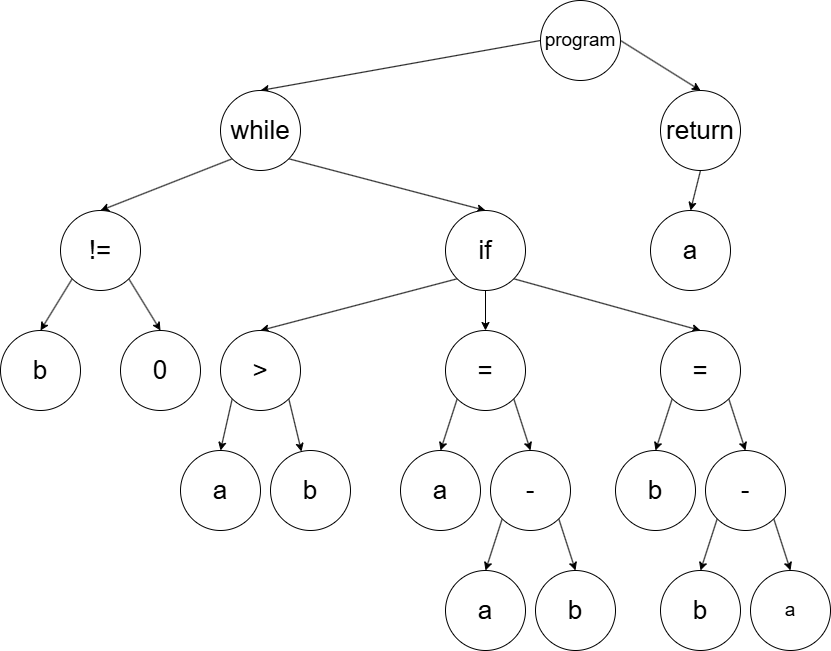
else:

b = b - a

return a

Should produce the following tree:





Please note that you need to create the node “program” by yourself.

**Requirements**

* You are not allowed to alter the provided classes.
* You can not use any additional libraries.
* You can not use the ast to solve this problem.
* You can not use any additional libraries.
* You must generate the tree from the ast.dump() function.
* You can not create the tree from the original source code.

## Problem 2 – Linked Binary Tree - 25 Points

Implement the member function \_swap(self,e, q) in the given linked binary tree class. The member function swaps the nodes e and q. Maintain child and parent references.

**Example**

|  |  |
| --- | --- |
| Swapping child and parent relationship | Swapping individual nodes |
|  |  |
|  |  |
| Swapping child and parent relationship | Swapping individual nodes |
|  |  |
|  |  |

**Requirements**

* You are not allowed to alter the provided classes.
* You are not allowed to just change node values.
* You are not allowed just to change node values.
* The time complexity requirement of this method is O(1).
* The space complexity requirement of this method is O(1).

## Problem 3 – Isomorphic - 25 Points

Implement the member is\_isomorphic(self, other), which analyzes whether 𝑠𝑒𝑙𝑓 and 𝑜𝑡ℎ𝑒𝑟 are isomorphic representations. If so, return true. Otherwise, return false. Isomorphic trees have one or more children flipped for nodes. Flips swap the left and right child of a node.

**Requirements**

* Your function has to be in O(n^2) time complexity.
* Your function has to be in O(n1) space complexity.
* You cannot use Python lists or any other built-in data structures.

## Problem 4 – One-Zero-Tree - 25 Points

Implement the function one\_zero\_update(tree) to update the tree's internal nodes based on the following rules. Please note that the function is not a class member:

* If a node has a depth of 0,2,4,6,.. its element equals the maximum of its children.
* If a node has a depth of 1,3,5,7,... its element equals the minimum of its children.

**Requirements**

* Your function has to be in O(n) time complexity.
* Your function has to return the same tree with all internal nodes updated.
* You cannot use Python lists or any other built-in data structures.