Report for 1DV516 - PA2

Algorithms and Advanced Data Structures

Ingmar Immanuel Falk

October 5, 2023

Contents

1 Introduction

2 Problem 5 - *Isomorphism*

To check if two trees are isomorphic we have to check if it is possible to obtain one of the trees by swapping the left and right subtrees of the other tree. This means, that we have to recursively iterate over every node of *both* trees to confirm either that the trees are isomorphic or that they are not.

The algorithm consists of only four parts:

- Check if both trees are empty. If so, they are isomorphic.
- Check if one of the trees is empty. If so, they are not isomorphic.
- Check if the values of nodes not are equal. If so, they are not isomorphic.
- Recursively check if either both the left and right side of both trees are isomorphic or if the left side of one tree is isomorphic with the right side of the other tree and vice versa.

2.1 Pseudocode

```
isIso(node1, node2):
    if (node1 is empty and node2 is empty) -> return true
    if (node1 is empty or node2 is empty) -> return false
    if (node1.value != node2.value) -> return false
    return (isIso(node1.left, node2.left) and isIso(node1.right, node2.right))
        or (isIso(node1.left, node2.right) and isIso(node1.right, node2.left))
```

3 Problem 6