

Data Structures  
Saving and Restoring Binary Trees

Cain Susko

Queen's University  
School of Computing

February 5, 2022

## Serialization

This is the process of generating a series of bits for a binary tree such that another computer could read these bits and restore the tree.

A possible way of doing this is using the order of a type of traversal. The problem with this is a reading order is not unique to a tree, however if we kept track of the number of nodes at each node as well as their traversal order. If one wrote down the order of a preorder traversal and the number of nodes' children sent it to someone else, they could redraw the tree if they knew how preorder traversals worked.

An algorithm for such a process is as follows

```
rebuildTree():
    /*read next (O,D) pair from file*/

    t = new tree(0)
    if ((D==2) || (D==L)):
        t.attachL(rebuildTree())
    if ((D==2) || (D==R)):
        t.attachR(rebuildTree())
    return t
```

Note: the code is not complete.

## Evaluation

Some trees represent structures that can be evaluated. If we wanted to evaluate an expression tree for example, we could use the following algorithm:

1. if the current node is a number, return its value
2. recursively evaluate the left subtree and get the result
3. recursively evaluate the right subtree and get the result
4. apply the operator in the current node to the left and right results, return the result.