Protect Trees
(Programming Club 4)

Tomasz Kosciuszko

22.10.2018

1 Introduction

Trees appear whenever we can observe some hierarchical structure. Think for example about a company ladder or directories on a disk or parsing sentences into grammatical forms. It is useful to know how to design some basic algorithms working for trees and in what form to represent them in memory.

2 How it works

Each tree consists of nodes and edges. It has a root (the most senior node), some inner nodes and the lowest are called leafs.

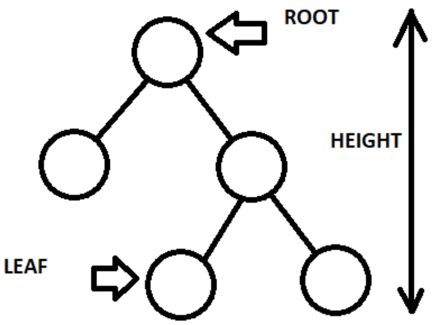

Figure 1: Small tree

It is good to represent the tree in a similar way as we did for linked lists a couple of weeks ago:

```
class Node {
    int value;
    vector <Node*> children;
};
class Tree {
    Node* root;
    int GetHeight();  // How would you implement these?
    void InsertValue(int a);
    vector <int> GetAllValues();
};
```

3 Try some coding!

Have a look here for interesting challenges of varying difficulty:
www.hackerrank.com/interview/interview-preparation-kit/trees/challenges
You can choose your favourite language and try to manipulate strings in interview-like problems.

4 Think about this

- How many edges can a tree with n nodes have?
- Somebody has drown x trees on the blackboard. If you count all nodes you get 99 and if you count all edges you get 77. How many trees are there?
- What is a binary tree?
- What is the minimum height of a binary tree with 10^{100} nodes? Maximum height?
- **(Challenge)** Given a tree find a node, such that if you make it the root then height of your tree is minimal. How would your algorithm work? Is there always one such node?