Exam

Jakub Dotlačil

December 19, 2018

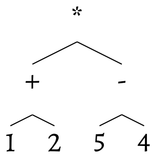
1 Assignment

In this exam, we will implement a tree structure, which is a hierarchical structure often used in computer science. You can use external packages, but you cannot use functions/classes that implement trees. If in doubt, you can ask me whether a particular function from a package is allowed.

A tree data structure can be defined recursively as a collection of nodes, where each node is a data structure consisting of a value, together with a list of references to other nodes (the *children*). A tree has a *root* node which is not referenced by any other node in the tree. Every other node in the tree is referenced exactly once. The trees that refer no other nodes are standardly called *leaves*.

A tree is a natural way to represent the structure of an expression. Unlike other notations, it can represent the computation unambiguously. For example, the computation (1 + 2) * (5 - 4) can be seen as a tree in which: (i) the nodes of an expression tree can be operands like 1 and 2 or operators like +, - and *; (ii) operands are leaf nodes; (iii) operator nodes contain references to their operands. It is common to draw the tree following the conventions as below (the root (the * operator in this case) is on top, the leaves (1, 2, 4, 5) are on the bottom):

(1)



We will first consider a binary tree (a tree in which every node has maximally two children). Later, we generalize the tree structure to arbitrary trees (any number of children is allowed.

In the exam, you will build a Tree class that implements the tree data structure and that allows operations on it. You can check the exam.py file for further instructions.