Tree Traversals in Rust Report

Simone Stanganini

October 19, 2023

This is a report of my implementation of the first hands-on of the course Competitive Programming and Contests.

- `fn check_bst()`: call the recursive function `rec_check_bst()`, this function checks recursive the left and right node, and they return the struct with one boolean value for the result, and both return the max value and the min value in their sub tree.

  Then we check if the current node respects the bst proprierties and we return the result.

- `rec_is_balanced()`: call the recursive function `rec_is_balanced()`, then we do a recursive call on the left and right node, the result is the height starting from the leaf, each recursive call adds 1, after we check if the height of the sub trees differs almost of 1.

- `is_maxheap()`: call the recursive function `rec_is_maxheap()`, with total = 0, where total is the variable responsible for the check of the completeness, we multiply total by 2, on each recursive call and add 1 for the left node and 2 for the right, because it's the number that we expect for the last right leaf, in fact if the total nodes are greater than the total variable, it means that the tree misses some nodes in the left so it doesn't respect the completeness properties.