Programming and Data Structures with Python 2026

Assignment 5

16 November 2026, due 26 November 2026

Lectures 24–25 describe how to implement a class `Tree` to maintain height balanced search trees. In this implementation, a tree is a collection of nodes, where each node stores `value` and `height` and is linked to its children other by `left` and `right`.

Write an alternative implementation of height balanced search trees in a class `DictTree` where each node is a dictionary with keys `value`, `left`, `right` and `height`.

Your class should implement the standard operations `find(v)`, `insert(v)` and `delete(v)`. You will also need to implement auxiliary functions as needed — for example, rotation and rebalancing. Printing a tree should list out the values in the tree in ascending order.

Instructions

1. Submit your final code as a single Python notebook.
2. Add documentation to explain at a high level what your code is doing.
3. Show some sample executions.