

## Zestaw 8: Struktury drzewiaste

Dana jest struktura drzewiasta zbudowana z węzłów:

```
class Node:
    def __init__(self):
        self.val = None
        self.left = None
        self.right = None
```

Proszę napisać następujące funkcje:

1. Funkcja wypisującą zawartość drzewa.
2. Funkcja, która sprawdza czy dana liczba należy do drzewa.
3. Funkcja, która zwróci rozmiar drzewa (liczbę węzłów).
4. Funkcja, która zwróci wysokość drzewa (ilość poziomów).
5. Funkcja, która zwróci liczbę liści drzewie.
6. Funkcja, która zwróci liczbę węzłów na n-tym poziomie.
7. Funkcja, która zwróci liczbę węzłów mających jednego potomka.
8. Funkcja, która usunie wszystkie węzły drzewa.
9. Funkcja, która usunie węzły powyżej n-tego poziomu.
10. Funkcja, która sprawdza czy dana liczba należy do drzewa BST.
11. Funkcja, która wstawi liczbę do drzewa BST.
12. Funkcja, która sprawdza czy dane drzewo jest poprawnym drzewem BST.

Każde z powyższych zadań można rozwiązać iteracyjnie albo rekurencyjnie.

13. Funkcja, która sprawdza czy dane drzewo BST jest poprawnym drzewem AVL.
14. Dana jest funkcja:

```
def f(a,b):
    if a==0: return b+1
    if b==0: return f(a-1,1)
    return f(a-1,f(a,b-1))
```

Proszę napisać funkcję w wersji iteracyjnej.

15. Dana jest funkcja rozwiązująca problem wież Hanoi:

```
def hanoi(n,a,b,c):
    if n>0:
        hanoi(n-1,a,c,b)
        print(a,'->',c)
        hanoi(n-1,b,a,c)
```

- Usunąć rekurencję zastępując ją stosem,
- Zaimplementować algorytm bez użycia stosu ani rekurencji (wikipedia).