

[2pkt.] Zadanie 1.

Szablon rozwiązania: zad1.py

Drzewo BST T reprezentowane jest przez obiekty klasy Node:

```
class Node:
    def __init__( self ):
        self.left    = None # lewe podrzewo
        self.right   = None # prawe podrzewo
        self.parent  = None # rodzic drzewa jeśli istnieje
        self.value   = None # przechowywana wartość
```

Proszę zaimplementować funkcję:

`ConvertTree(T)`

która przekształca drzewo T na drzewo o minimalnej wysokości, w którym węzły spełniają warunek: największy element na danym poziomie jest mniejszy od najmniejszego elementu na kolejnym poziomie. Funkcja zwraca korzeń nowego drzewa. Poziomy numerujemy od korzenia do liści.

Funkcja powinna być możliwie jak najszybsza oraz—jako kryterium drugiego rzędu—używać jak najmniejszej ilości pamięci (poza pamięcią już wykorzystaną na reprezentację drzewa). Proszę oszacować złożoność czasową oraz pamięciową użytego algorytmu.

Przykład poprawnego przekształcenia.

