**Лабораторна робота №4**

Візуалізація бінарного дерева

1. Створити бінарне дерево за номером студента та виконати його візуалізацію за допомогою алгоритму Д. Кнута.



Рішення:

import random

import matplotlib.pyplot as plt

import networkx as nx

class Node:

def \_\_init\_\_(self, key):

self.key = key

self.left = None

self.right = None

self.parent = None

class Tree:

def \_\_init\_\_(self):

self.root = None

def add\_node(self, key, node=None):

if node is None:

node = self.root

if self.root is None:

self.root = Node(key)

else:

if key <= node.key:

if node.left is None:

node.left = Node(key)

node.left.parent = node

else:

self.add\_node(key, node=node.left)

else:

if node.right is None:

node.right = Node(key)

node.right.parent = node

else:

self.add\_node(key, node=node.right)

def tree\_data(self, node=None):

if node is None:

node = self.root

stack = []

while stack or node:

if node is not None:

stack.append(node)

node = node.left

else:

node = stack.pop()

yield node.key

node = node.right

def add\_edges(graph, node, pos, x=0, y=0, layer=1):

if node is not None:

pos[node.key] = (x, y)

if node.left:

graph.add\_edge(node.key, node.left.key)

l = x - 1 / layer

add\_edges(graph, node.left, pos, x=l, y=y-1, layer=layer+1)

if node.right:

graph.add\_edge(node.key, node.right.key)

r = x + 1 / layer

add\_edges(graph, node.right, pos, x=r, y=y-1, layer=layer+1)

def plot\_tree(tree):

graph = nx.DiGraph()

pos = {}

add\_edges(graph, tree.root, pos)

nx.draw(graph, pos, with\_labels=True, arrows=False)

plt.show()

t = Tree()

for \_ in range(15):

t.add\_node(random.randint(-50, 50))

plot\_tree(t)

Зображення, що містить ряд, коло, мистецтво

Автоматично згенерований опис

Виконала: Павленко Д.А.

Група: ІКМ-223в