**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 5 з дисципліни

«Проектування алгоритмів»

„**Проектування і аналіз алгоритмів для вирішення NP-складних задач ч.2**”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-12 Казаков Володимир Олексійович*

**Перевірив**

(прізвище, ім'я, по батькові)

*Головченко М.Н.*

Київ 2022

Покроковий алгоритм:

1. Згенерувати матрицю розмірністю 300\*300 з випадковими значеннями відстаней між точками

2. Згенерувати порожню матрицю, у якій пізніше будуть зберігатися значення відстаней, що пролетіла кожна бджола

3. Згенерувати порожню матрицю для зберігання шляху кожної бджоли

4. Кожна бджола в циклі знаходить випадкові точки з усіх 300, доки врешті-решт не залишиться непройдених точок

5. Обчислення відстані, що пройшла кожна бджола. Бджола з найменшою відстанню і є результатом роботи програми

Вихідний код (Python):

import random

#генерує матрицю з усіма відстанями між точками

def generate\_table(nodes\_amount):

matrix = []

for i in range(nodes\_amount):

matrix.append([])

for j in range(nodes\_amount):

matrix[i].append(random.randint(5, 150))

return matrix

#генерує матрицю з відстанями між кожними двома точками, між якими летіла бджола

def generate\_bees():

bees = []

for i in range(20):

bees.append([])

return bees

#генерує матрицю зі шляхом з точок, через які пролетіла бджола

def generate\_bee\_ways():

bee\_ways = []

for i in range(20):

bee\_ways.append([])

bee\_ways[i].append(0)

return bee\_ways

#обчислює повну відстань, пройдену бджолою

def calculate\_distance(bee):

distance = 0

for node in bee:

distance = distance + node

return distance

distance\_matrix = generate\_table(300)

bees = generate\_bees()

bee\_ways = generate\_bee\_ways()

count = 0

for bee in bees:

current\_node = 0

next\_node = 0

uncompleted\_nodes = []

k = 1

while k < 300:

uncompleted\_nodes.append(k)

k = k + 1

for i in range(299):

next\_node = random.choice(uncompleted\_nodes)

bee.append(distance\_matrix[current\_node][next\_node])

uncompleted\_nodes.remove(next\_node)

bee\_ways[count].append(next\_node)

current\_node = next\_node

count = count + 1

best\_bee = 0

count = 0

for bee in bees:

if calculate\_distance(bee) < calculate\_distance(bees[best\_bee]):

best\_bee = count

count = count + 1

print("The best bee is bee №",best\_bee)

print("Distance =", calculate\_distance(bees[best\_bee]))

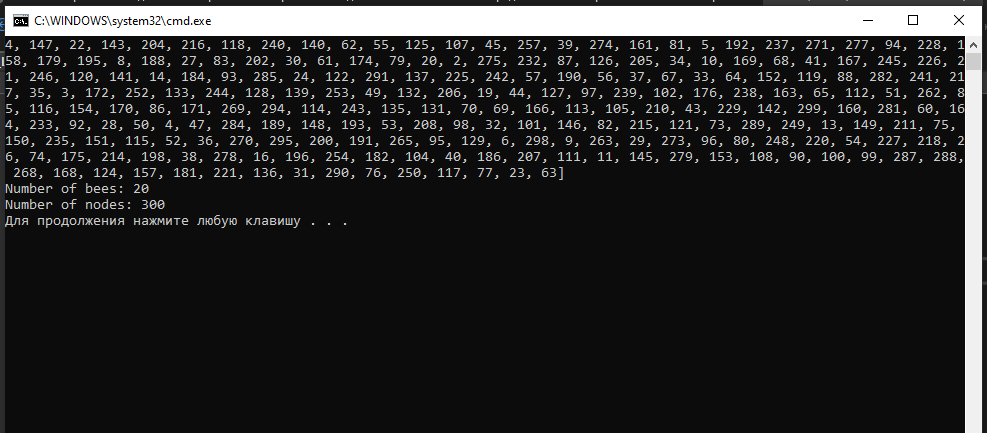
print("Way:")

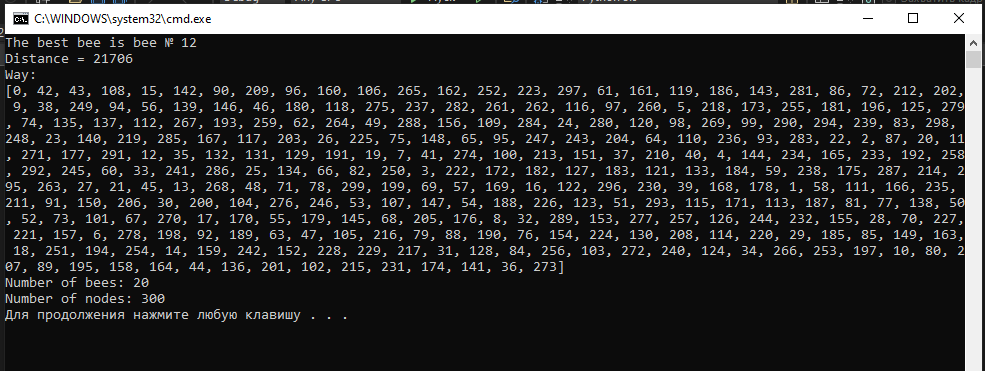
print(bee\_ways[best\_bee])

print("Number of bees:", 20)

print("Number of nodes:", 300)

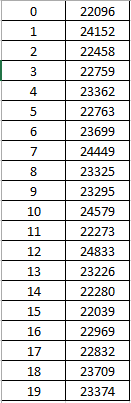
Приклади роботи програми:





Тестування алгоритму:

Вивід пройдених відстаней кожної бджоли:



Перша колонка – номери бджіл, друга колонка – пройдені відстані.

Висновок: Розроблено та протестовано програму, що розв’язує програму за допомогою бджолиного алгоритму