

## ДОСЛІДЖЕННЯ МУРАШИНИХ АЛГОРИТМІВ

### Хід роботи:

18	Тернопіль	232	877	1100	298	977	134	427	547	1289	159	128	754	660	825	162	1097
----	-----------	-----	-----	------	-----	-----	-----	-----	-----	------	-----	-----	-----	-----	-----	-----	------

					ДУ «Житомирська політехніка».20.121.18.						
Змн.	Арк.	№ докум.	Підпис	Дата					Літ.	Арк.	Аркушів
Розроб.		Соболевський Д.А.									
Перевір.		Філіпов В.О.								1	4
Керівник									ФІКТ Гр. ІПЗк-20-1		
Н. контр.											
Зав. каф.											

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 class CityMap:
5     def __init__(self, distancesMatrix, number_of_cities):
6         self.distances = distancesMatrix
7         self.number_of_cities = number_of_cities
8         self.pheromones = [[np.random.rand() for j in range(number_of_cities)] for i in range(number_of_cities)]
9
10    def UpdatePheromones(self, evaporationRate, pheromoneDelta):
11        for i, row in enumerate(self.pheromones):
12            for j, col in enumerate(row):
13                self.pheromones[i][j] *= (1 - evaporationRate)
14                self.pheromones[i][j] += pheromoneDelta[i][j]
15
16    class Ant:
17        def __init__(self, starting_city):
18            self.starting_city = starting_city
19            self.current_city = starting_city
20            self.distance = 0
21            self.visited_cities = [starting_city]
22
23        def Move(self, new_city, distance):
24            self.current_city = new_city
25
26            self.visited_cities.append(new_city)
27
28            self.distance += distance
29
30    class Colony:
31        max_colony_cycles = 50
32        pheromone_addition = 0.0005
33        pheromone_evaporation_rate = 0.2
34        pheromone_importance = 0.01
35        distance_importance = 9.5
36        ant_can_visit_previous_cities = False
37
38        def __init__(self, number_of_ants):
39            self.number_of_ants = number_of_ants
40
41        # Пошук найкоротшого шляху
42        def FindRoute(self, city_map, city_number):
43            min_distance = float('inf')
44            route = []
45
46            for cycle in range(self.max_colony_cycles):
47                pheromones_delta = [[0.0 for i in range(city_map.number_of_cities)] for j in range(city_map.number_of_cities)]
48
49                for ant_number in range(self.number_of_ants):
50                    ant = Ant(city_number)
51
52                    while len(ant.visited_cities) < city_map.number_of_cities:
53                        next_city = self.GetNextCity(ant, city_map)
54                        ant.Move(next_city, city_map.distances[ant.current_city][next_city])
55
56                        ant_distance = ant.distance + city_map.distances[ant.current_city][ant.starting_city]
57
58                        if ant_distance < min_distance:
59                            min_distance = ant_distance
60                            route = ant.visited_cities
61
62                    for city in range(len(ant.visited_cities) - 1):
63                        pheromones_delta[ant.visited_cities[city]][ant.visited_cities[city + 1]] += self.pheromone_addition / ant_distance
64
65                city_map.UpdatePheromones(self.pheromone_evaporation_rate, pheromones_delta)
66
67            return min_distance, route
68

```

		Соболевський Д.А.			ДУ «Житомирська політехніка».20.121.18 – Лр5	Арк.
		Філіпов В.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

69 # Формування списку ймовірностей переміщення в місто для мурахи
70 def GetProbabilities(self, ant, city_map):
71     result = [0 for i in range(city_map.number_of_cities)]
72     total_probability = 0
73
74     for new_city in range(city_map.number_of_cities):
75         if (new_city != ant.current_city) and (self.ant.can_visit_previous_cities or new_city not in ant.visited_cities):
76             probability = pow(city_map.pheromones[ant.current_city][new_city], self.pheromone_importance) * pow(1 / city_map.distances[ant.current_
77             result[new_city] = probability
78             total_probability += probability
79
80     result = [result[i] / total_probability for i in range(city_map.number_of_cities)]
81
82     return result
83
84 # Вибір наступного міста для мурахи
85 def GetNextCity(self, ant, city_map):
86     probabilities = self.GetProbabilities(ant, city_map)
87     random_value = np.random.rand()
88
89     for i in range(city_map.number_of_cities):
90         if probabilities[i] > random_value:
91             return i
92         else:
93             random_value -= probabilities[i]
94
95     return -1
96
97 # Відстані між містами
98 distance = [
99     [0, 645, 868, 125, 748, 366, 256, 316, 1057, 382, 360, 471, 428, 593, 311, 844, 602, 232, 575, 734, 521, 120, 343, 312, 396],
100    [645, 0, 252, 664, 81, 901, 533, 294, 394, 805, 975, 343, 468, 196, 957, 446, 430, 877, 1130, 213, 376, 765, 324, 891, 672],
101    [868, 252, 0, 858, 217, 1171, 727, 520, 148, 1111, 1221, 611, 731, 390, 1045, 591, 706, 1100, 1391, 335, 560, 988, 547, 1141, 867],
102    [125, 664, 858, 0, 738, 431, 131, 407, 1182, 257, 423, 677, 557, 468, 187, 803, 477, 298, 671, 690, 624, 185, 321, 389, 271],
103    [748, 81, 217, 738, 0, 1119, 607, 303, 365, 681, 833, 377, 497, 270, 925, 365, 477, 977, 1488, 287, 297, 875, 405, 957, 747],
104    [366, 901, 1171, 431, 1119, 0, 561, 618, 1402, 328, 135, 747, 627, 898, 296, 1070, 908, 134, 280, 1040, 798, 246, 709, 143, 701],
105    [256, 533, 727, 131, 607, 561, 0, 298, 811, 388, 550, 490, 489, 337, 318, 972, 346, 427, 806, 478, 551, 315, 190, 538, 149],
106    [316, 294, 520, 407, 303, 618, 298, 0, 668, 664, 710, 174, 294, 246, 627, 570, 506, 547, 883, 387, 225, 435, 126, 637, 363],
107    [1057, 394, 148, 1182, 365, 1402, 811, 668, 0, 1199, 1379, 857, 977, 474, 1129, 739, 253, 1289, 1539, 333, 806, 1177, 706, 1292, 951],
108    [382, 805, 1111, 257, 681, 328, 388, 664, 1199, 0, 152, 780, 856, 725, 70, 1052, 734, 159, 413, 866, 869, 263, 578, 336, 949],
109    [360, 975, 1221, 423, 833, 135, 550, 710, 1379, 152, 0, 850, 970, 891, 232, 1173, 896, 128, 261, 1028, 1141, 240, 740, 278, 690],
110    [471, 343, 611, 677, 377, 747, 490, 174, 857, 780, 850, 0, 120, 420, 864, 282, 681, 754, 999, 556, 51, 590, 300, 642, 640],
111    [428, 468, 731, 557, 497, 627, 489, 294, 977, 856, 970, 120, 0, 540, 741, 392, 800, 660, 1009, 831, 171, 548, 420, 515, 529],
112    [593, 196, 390, 468, 270, 898, 337, 246, 474, 725, 891, 420, 540, 0, 665, 635, 261, 825, 1149, 141, 471, 653, 279, 892, 477],
113    [311, 957, 1045, 187, 925, 296, 318, 627, 1129, 70, 232, 864, 741, 665, 0, 1157, 664, 162, 484, 805, 834, 193, 508, 331, 458],
114    [844, 446, 591, 803, 365, 1070, 972, 570, 739, 1052, 1173, 282, 392, 635, 1157, 0, 896, 1097, 1363, 652, 221, 964, 696, 981, 1112],
115    [602, 430, 706, 477, 477, 908, 346, 506, 253, 734, 896, 681, 800, 261, 664, 896, 0, 774, 1138, 190, 732, 662, 540, 883, 350],
116    [232, 877, 1100, 298, 977, 134, 427, 547, 1289, 159, 128, 754, 660, 825, 162, 1097, 774, 0, 338, 987, 831, 112, 575, 176, 568],
117    [575, 1130, 1391, 671, 1488, 280, 806, 883, 1539, 413, 261, 999, 1009, 1149, 484, 1363, 1138, 338, 0, 1299, 1065, 455, 984, 444, 951],
118    [734, 213, 335, 690, 287, 1040, 478, 387, 333, 866, 1028, 556, 831, 141, 805, 652, 190, 987, 1299, 0, 576, 854, 420, 1036, 608],
119    [521, 376, 560, 624, 297, 798, 551, 225, 806, 869, 1141, 51, 171, 471, 834, 221, 732, 831, 1065, 576, 0, 641, 351, 713, 691],
120    [120, 765, 988, 185, 875, 246, 315, 435, 1177, 263, 240, 590, 548, 653, 193, 964, 662, 112, 455, 854, 641, 0, 463, 190, 455],
121    [343, 324, 547, 321, 405, 709, 190, 126, 706, 578, 740, 300, 420, 279, 508, 696, 540, 575, 984, 420, 351, 463, 0, 660, 330],
122    [312, 891, 1141, 389, 957, 143, 538, 637, 1292, 336, 278, 642, 515, 892, 331, 981, 883, 176, 444, 1036, 713, 190, 660, 0, 695],
123    [396, 672, 867, 271, 747, 701, 149, 363, 951, 949, 690, 640, 529, 477, 458, 1112, 350, 568, 951, 608, 691, 455, 330, 695, 0]
124 ]
125
126 # Список міст
127 cities = [
128     'Вінниця', 'Дніпро', 'Донецьк', 'Житомир', 'Запоріжжя', 'Івано-Франківськ', 'Київ', 'Кропивницький',
129     'Луганськ', 'Луцьк', 'Львів', 'Миколаїв', 'Одеса', 'Полтава', 'Рівне', 'Сімферополь', 'Суми', 'Тернопіль',
130     'Ужгород', 'Харків', 'Херсон', 'Хмельницький', 'Черкаси', 'Чернівці', 'Чернівів'
131 ]
132
133 # Пошук відповіді задачі
134 city_map = CityMap(distance, len(distance[0]))
135 colony = Colony(len(distance[0]))
136 result = colony.FindRoute(city_map, 18)
137
138 print(f"Отриманий найкоротший шлях: {result[0]} км")
139
140 # Вивід отриманого маршруту
141 city_routes = "Отриманий маршрут: "
142
143 for i in result[1]:
144     city_routes += cities[i]
145     if i != result[1][-1]:
146         city_routes += "->"
147
148 print(city_routes)
149
150 # Графічне відображення отриманих даних
151 fig = plt.figure(figsize=(13, 13))
152 plt.xticks([i + 1 for i in range(25)])
153 plt.yticks([i for i in range(25)], cities)
154 plt.xlabel("Номери міст")
155 plt.ylabel("Назви міст")
156 plt.title("Маршрут, пройдений комівояжером")
157 plt.plot([i + 1 for i in range(25)], result[1], ms = 10, marker = 'o', mfc = 'r')
158 plt.grid()
159 plt.show()

```

Рис. 1. Код файлу LR\_7\_task\_1

		Соболевський Д.А.			ДУ «Житомирська політехніка».20.121.18 – Лр5	Арк.
		Філіпов В.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

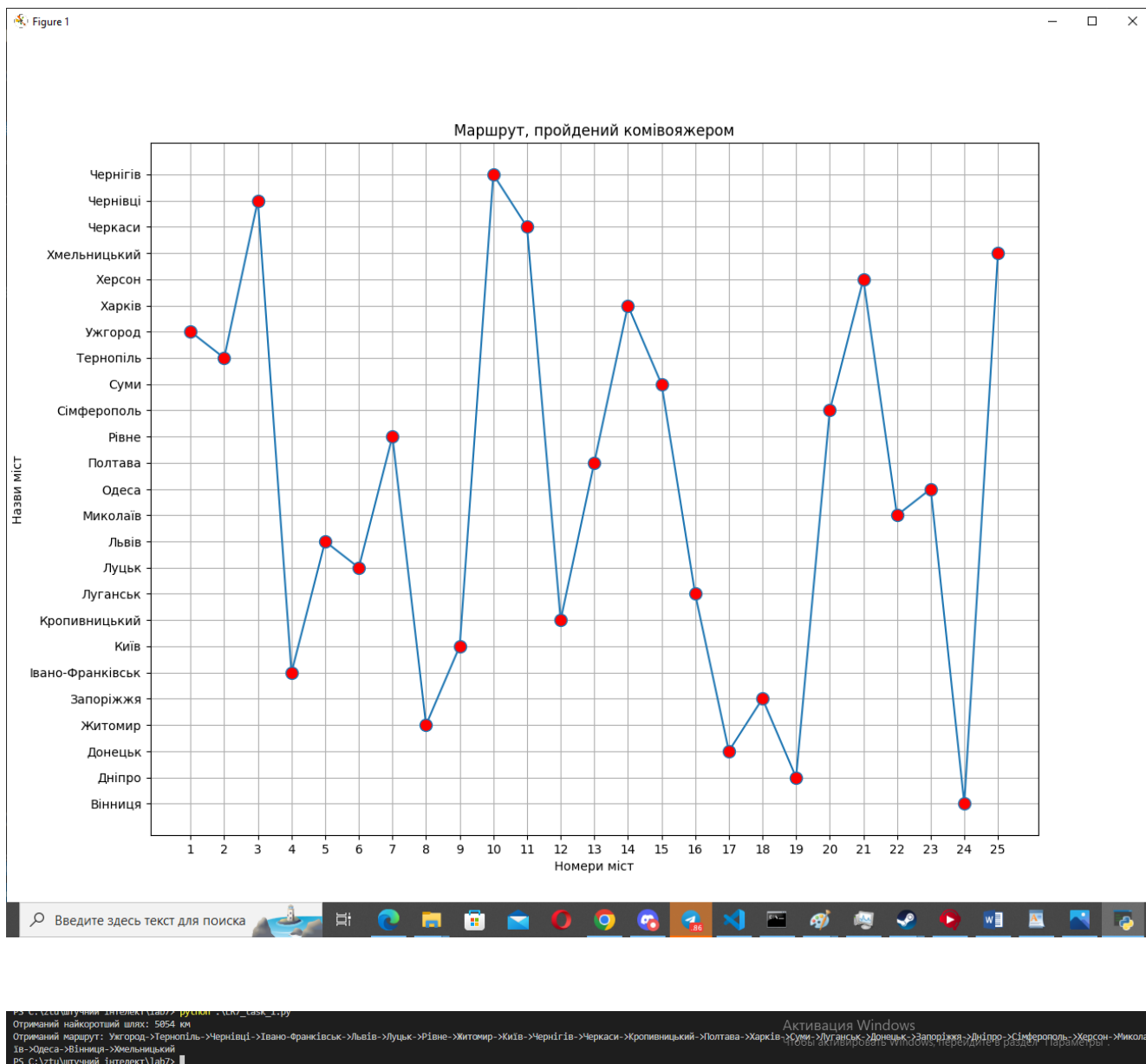


Рис. 2. Результат програми

Програма розрахувала найкоротший шлях, він склав 5054 кілометрів. Подорож розпочалась з Ужгорода, що відповідає місту, вказаному в завданні. Шлях подорожі продемонстрований на графіку. Для проведення дослідження були використані відкриті дані відстаней між обласними центрами України.

Висновок: в данній лабораторній роботі навчився використовувати спеціалізовані бібліотеки та мову програмування Python, навчився досліджувати метод мурашиних колоній.

		Соболевський Д.А.			ДУ «Житомирська політехніка».20.121.18 – Лр5	Арк. 4
		Філіпов В.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		