



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по Лабораторной работе №6

по курсу «Анализ Алгоритмов»

на тему: «Методы решения задачи коммивояжера»

Студент группы ИУ7-56Б

(Подпись, дата)

Мансуров В. М.

(Фамилия И.О.)

Преподаватель

(Подпись, дата)

Волкова Л. Л.

(Фамилия И.О.)

Преподаватель

(Подпись, дата)

Строганов Ю. В..

(Фамилия И.О.)

Москва — 2023 г.

Содержание

Введение	3
1 Аналитическая часть	4
1.1 Метод на основе муравьиного алгоритма	4
2 Конструкторская часть	7
2.1 Требования к программному обеспечению	7
2.2 Разработка алгоритмов	7
2.3 Описание используемых типов данных	14
3 Технологическая часть	15
3.1 Средства реализации	15
3.2 Сведения о модулях программы	15
3.3 Реализация алгоритмов	15
3.4 Функциональные тесты	20
4 Исследовательская часть	21
4.1 Технические характеристики	21
4.2 Демонстрация работы программы	21
4.3 Временные характеристики	22
4.4 Постановка эксперимента	24
4.4.1 Класс данных 1	24
4.4.2 Класс данных 2	29
4.5 Вывод	32
Заключение	34
Список использованных источников	35
Приложение А	36
Приложение Б	54

Введение

Оптимизации, позволяющие улучшить работу существующих алгоритмов или помогающие решить поставленную задачу иным, более эффективным способом, были важны во все времена. Одной из важных задач являются задачи поисков оптимальных маршрутов.

Целью данной лабораторной работы является параметризация метода решения задачи коммивояжера на основе муравьиного метода.

Для поставленной цели необходимо выполнить следующие задачи.

- 1) Описать задачу коммивояжера.
- 2) Описать методы решения задачи коммивояжера — метод полного перебора и метод на основе муравьиного алгоритма.
- 3) Привести схемы муравьиного алгоритма и алгоритма, позволяющего решить задачу коммивояжера методом полного перебора.
- 4) Разработать и реализовать программный продукт, позволяющий решить задачу коммивояжера исследуемыми методами.
- 5) Сравнить по времени метод полного перебора и метод на основе муравьиного алгоритма.
- 6) Описать и обосновать полученные результаты в отчете о выполненной лабораторной работе.

Выданный индивидуальный вариант для выполнения лабораторной работы:

- неориентированы граф;
- без элитных муравьев;
- гамильтонов цикл;
- 80 дней вокруг света.

1 Аналитическая часть

В этом разделе будет представлена информация о задаче коммивояжера, а также о способах её решения — методе полного перебора и методе на основе муравьиного алгоритма.

Задача коммивояжера (англ. *traveling salesman problem*) — (задача о бродячем торговце) одна из самых важных задач всей транспортной логистики, в которой рассматриваются вершины графа, а также матрица смежности (для расстояния между вершинами) [1]. Задача заключается в том, чтобы найти такой порядок посещения вершин графа, при котором путь будет минимален, каждая вершина будет посещена лишь один раз, а возврат произойдет в начальную вершину.

Полный перебор для задачи коммивояжера [2] имеет высокую сложность алгоритма ($n!$), где n — количество городов. Суть в полном переборе всех возможных путей в графе и выбор наименьшего из них. Решение будет получено, но имеются большие затраты по времени выполнения при уже небольшом количестве вершин в графе.

1.1 Метод на основе муравьиного алгоритма

Муравьиный алгоритм (англ. *ant colony optimization*) [2] — метод решения задачи оптимизации, основанный на принципе поведения колонии муравьев.

Муравьи действуют, руководствуясь органами чувств. Каждый муравей оставляет на своем пути феромоны, чтобы другие могли ориентироваться. При большом количестве муравьев наибольшее количество феромона остается на наиболее посещаемом пути, посещаемость же может быть связана с длинами ребер.

Суть в том, что отдельно взятый муравей мало что может, поскольку он способен выполнять только максимально простые задачи. Но при большом числе других таких муравьев они могут выступать самостоятельными вычислительными единицами. Муравьи используют непрямой обмен

информацией через окружающую среду посредством феромона.

Пусть муравей имеет следующие характеристики:

- 1) зрение — способность определить длину ребра;
- 2) память — способность запомнить пройденный маршрут;
- 3) обоняние — способность чують феромон.

Также введем целевую функцию (1.1), характеризующую привлекательность ребра, определяемую благодаря зрению.

$$\eta_{ij} = 1/D_{ij}, \quad (1.1)$$

где D_{ij} — расстояние от текущего пункта i до заданного пункта j .

Также понадобится формула вычисления вероятности перехода в заданную точку (1.2).

$$p_{k,ij} = \begin{cases} \frac{\eta_{ij}^\alpha \cdot \tau_{ij}^\beta}{\sum_{q \notin J_k} \eta_{iq}^\alpha \cdot \tau_{iq}^\beta}, j \notin J_k \\ 0, j \in J_k \end{cases} \quad (1.2)$$

где a — параметр влияния длины пути, b — параметр влияния феромона, τ_{ij} — количество феромонов на ребре ij , η_{ij} — привлекательность ребра ij , J_k — список посещенных за текущий день городов.

После завершения движения всех муравьев (ночью, перед наступлением следующего дня), феромон обновляется по формуле (1.3).

$$\tau_{ij}(t+1) = \tau_{ij}(t) \cdot (1-p) + \Delta\tau_{ij}(t). \quad (1.3)$$

При этом

$$\Delta\tau_{ij}(t) = \sum_{k=1}^N \Delta\tau_{ij}^k(t), \quad (1.4)$$

где

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, \text{ ребро посещено муравьем } k \text{ в текущий день } t, \\ 0, \text{ иначе} \end{cases} \quad (1.5)$$

Поскольку вероятность (1.2) перехода в заданную точку не должна

быть равна нулю, необходимо обеспечить неравенство $\tau_{ij}(t)$ нулю посредством введения дополнительного минимально возможного значения феромона τ_{min} и в случае, если $\tau_{ij}(t + 1)$ принимает значение, меньшее τ_{min} , откатывать феромон до этой величины.

Путь выбирается по следующей схеме.

- 1) Каждый муравей имеет список запретов — список уже посещенных городов (вершин графа).
- 2) Муравьиное зрение отвечает за эвристическое желание посетить вершину.
- 3) Муравьиное обоняние отвечает за ощущение феромона на определенном пути (ребре). При этом количество феромона на пути (ребре) в день t обозначается как $\tau_{i,j}(t)$.
- 4) После прохождения определенного ребра муравей откладывает на нем некоторое количество феромона, которое показывает оптимальность сделанного выбора, это количество вычисляется по формуле (1.5).

Вывод

В данном разделе была рассмотрена задача коммивояжера, а также способы её решения — полный перебор и муравьиный алгоритм.

2 Конструкторская часть

В данном разделе будут представлены схемы алгоритма полного перебора и муравьиного алгоритма.

2.1 Требования к программному обеспечению

К программе предъявлены ряд требований:

- программа должна получать на вход матрицу смежности, для которой можно будет выбрать один из алгоритмов поиска оптимальных путей (полным перебором или муравьиным алгоритмом);
- программа должна позволять пользователю определять коэффициенты и количество дней для муравьиного алгоритма;
- программа должна давать возможность получить минимальную сумму пути, а также сам путь, используя один из алгоритмов.

2.2 Разработка алгоритмов

На рисунке 2.1 представлена схема алгоритма полного перебора путей. На рисунках 2.2 и 2.3 представлена схема алгоритма муравьиного алгоритма.

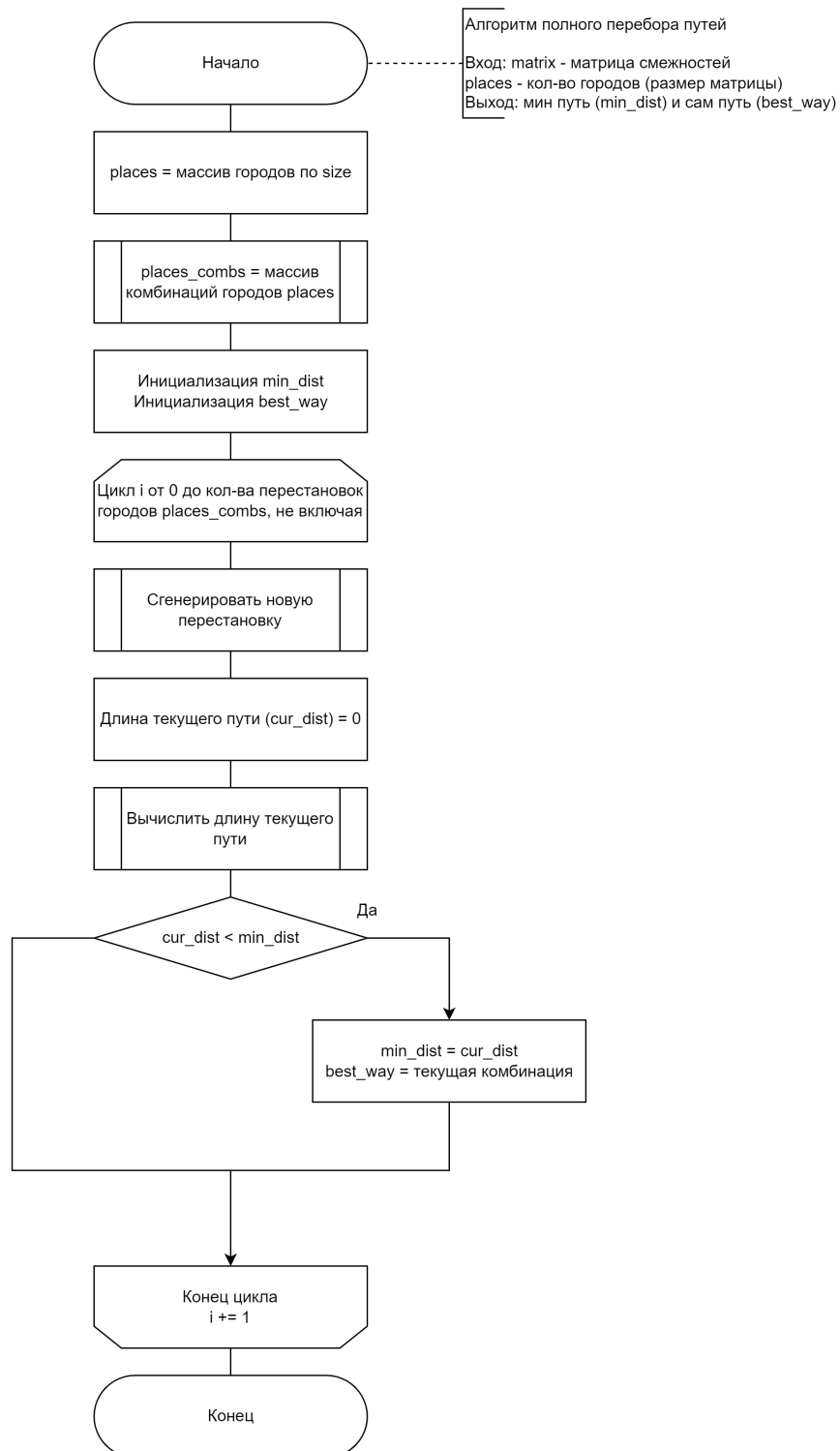


Рисунок 2.1 – Схема алгоритма полного перебора путей

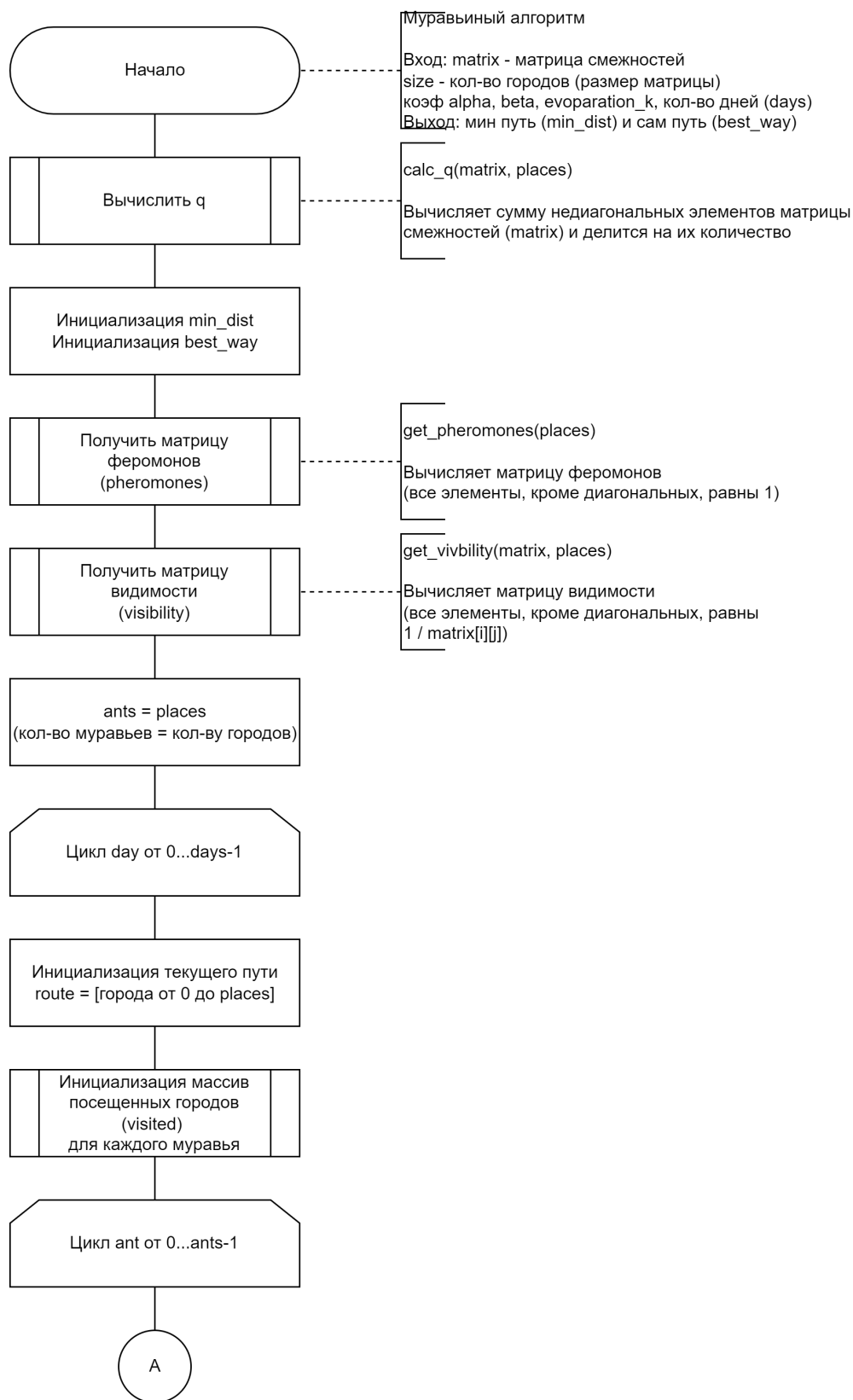


Рисунок 2.2 – Схема муравьиного алгоритма (часть 1)

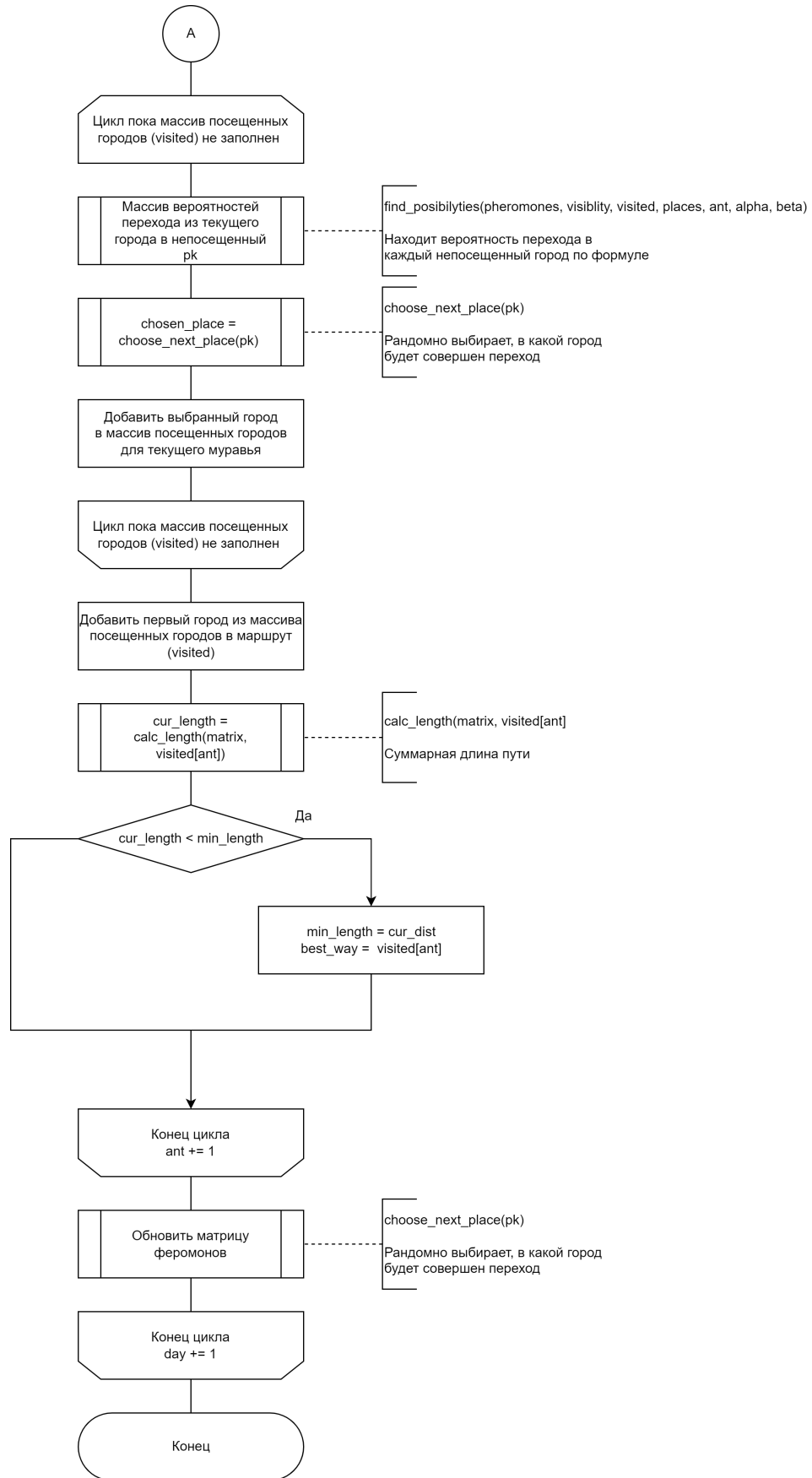


Рисунок 2.3 – Схема муравьиного алгоритма (часть 2)

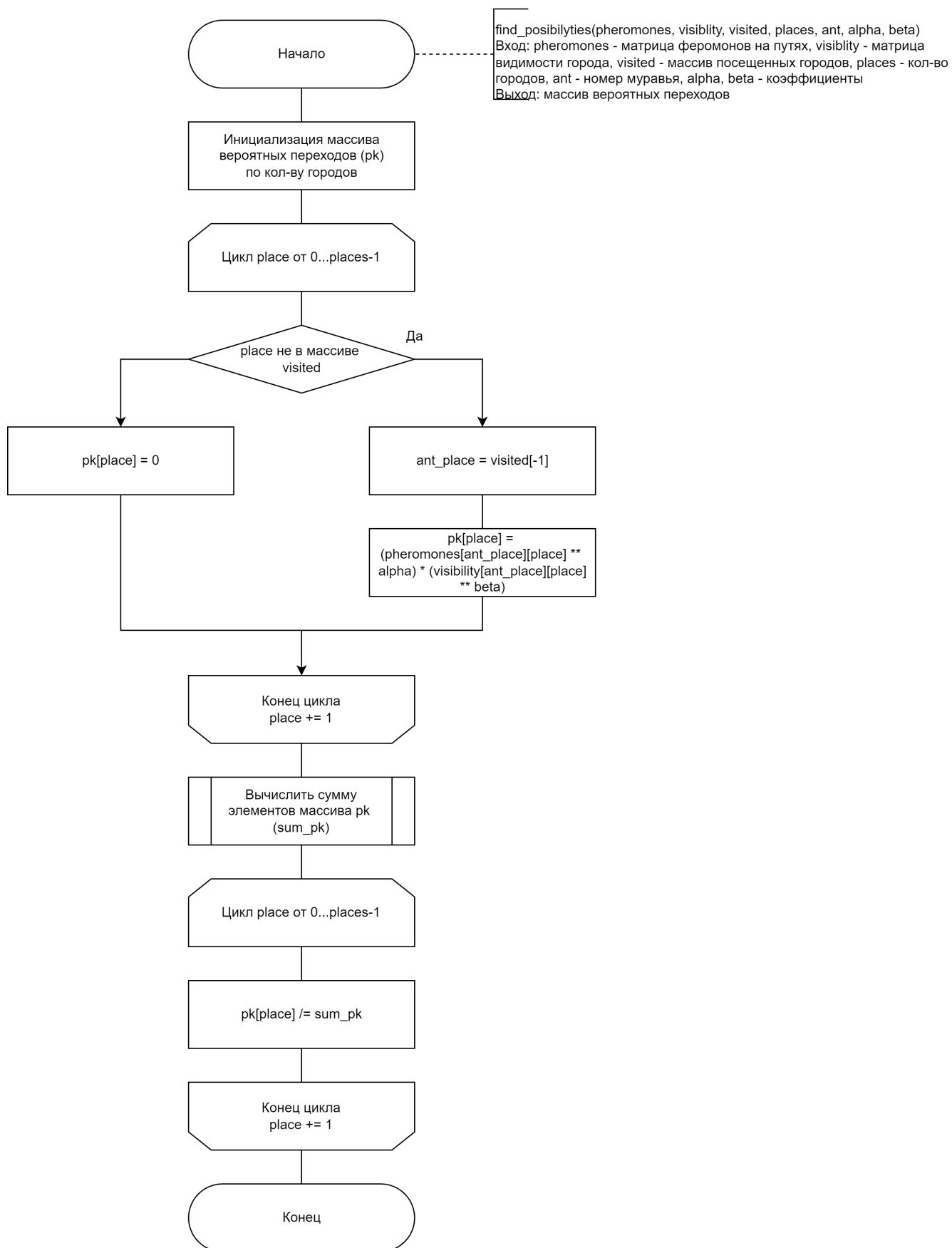


Рисунок 2.4 – Схема алгоритма нахождения массива вероятностей переходов в непосещенные города

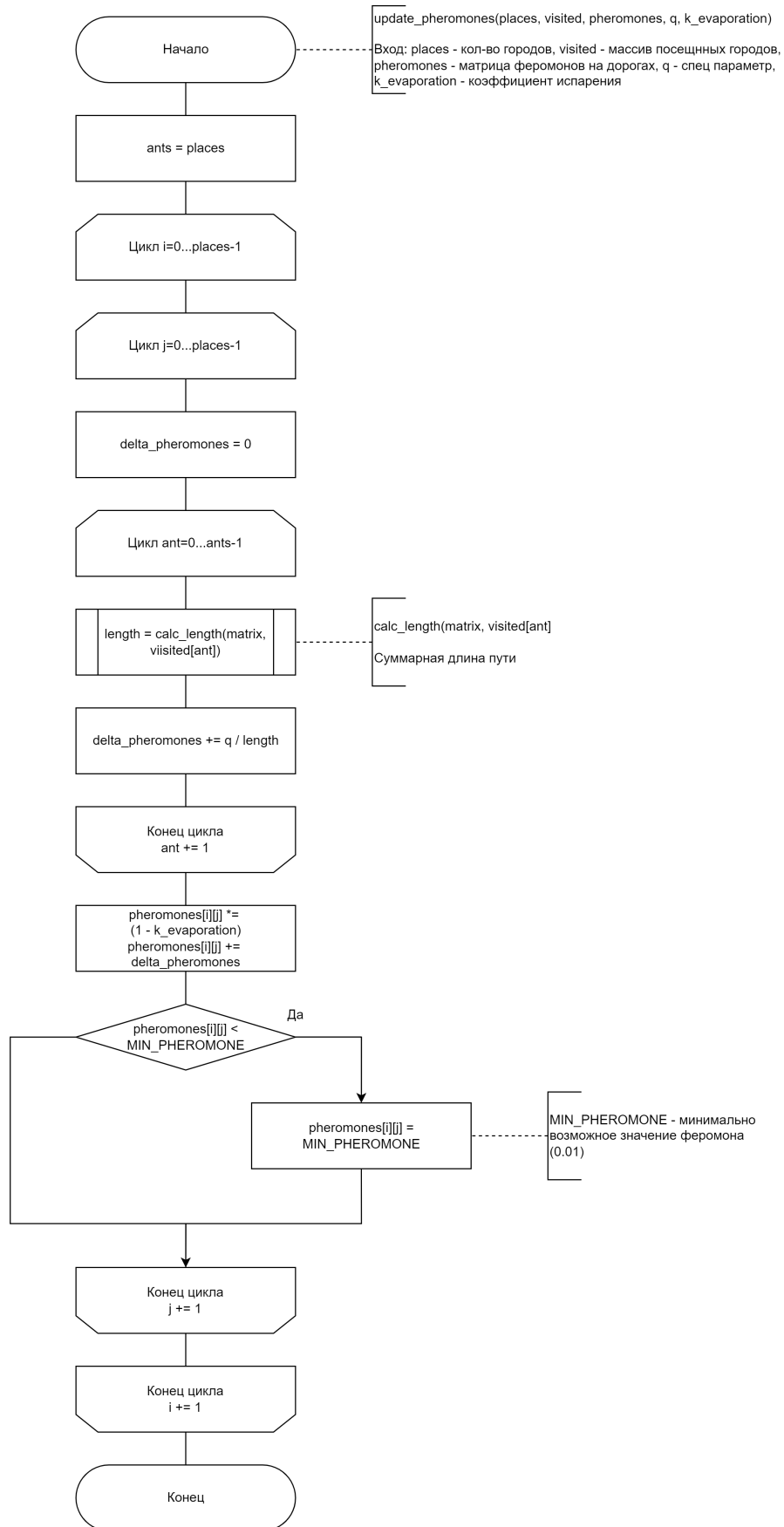


Рисунок 2.5 – Схема алгоритма обновления матрицы феромонов

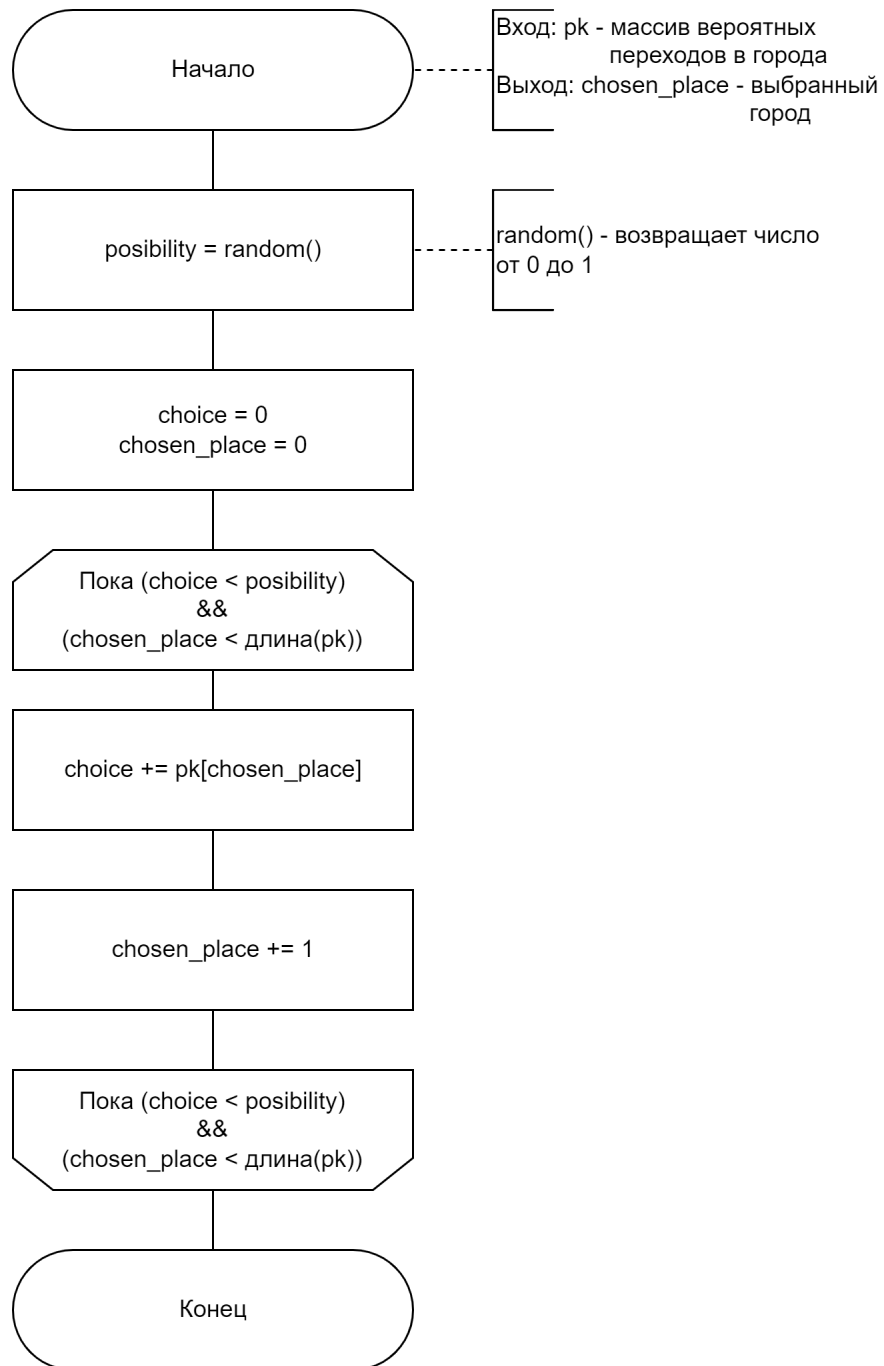


Рисунок 2.6 – Схема алгоритма выбора следующего города

2.3 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

- размер матрицы смежности — целое число;
- имя файла — строка;
- коэффициенты $\alpha, \beta, k_{evaporation}$ — действительные числа;
- матрица смежности — матрица целых чисел.

Вывод

В данном разделе была рассмотрена задача коммивояжера, а также полный перебор для её решения и муравьиный алгоритм. Были представлены требования к разрабатываемому программному обеспечению.

3 Технологическая часть

В данном разделе рассмотрены средства реализации, а также представлены листинги реализаций алгоритма расчета термовой частота для всех термов из выборки документов.

3.1 Средства реализации

В данной работе для реализации был выбран язык программирования *Python* [3]. В текущей лабораторной работе требуется замерить процессорное время работы выполняемой программы. Инструменты для этого присутствуют в выбранном языке программирования.

Время работы было замерено с помощью функции *process_time(...)* из библиотеки *time* [4].

3.2 Сведения о модулях программы

Данная программа разбита на следующие модули:

- *main.py* — файл, содержащий точку входа;
- *menu.py* — файл, содержащий код меню программы;
- *utils.py* — файл, содержащий служебные алгоритмы;
- *constants.py* — файл, содержащий константы программы;
- *algorithms.py* — файл, содержащий код всех алгоритмов.

3.3 Реализация алгоритмов

В листинге 3.1 представлен реализация алгоритм полного перебора путей, а в листингах 3.2–3.6 — муравьиный алгоритм и дополнительные к нему функции.

Листинг 3.1 – Реализация алгоритма полного перебора

```
1 def fullCombinationAlg(matrix, size):
2     places = np.arange(size)
3     placesCombinations = list()
4
5     for combination in itertools.permutations(places):
6         combArr = list(combination)
7         placesCombinations.append(combArr)
8
9     minDist = float("inf")
10
11    for i in range(len(placesCombinations)):
12        placesCombinations[i].append(placesCombinations[i][0])
13        curDist = 0
14        for j in range(size):
15            startCity = placesCombinations[i][j]
16            endCity = placesCombinations[i][j + 1]
17            curDist += matrix[startCity][endCity]
18
19        if (curDist < minDist):
20            minDist = curDist
21            bestWay = placesCombinations[i]
22
23    return minDist, bestWay
```


Листинг 3.2 – Реализация муравьиного алгоритма

```
1     q = calcQ(matrix, places)
2     bestWay = []
3     minDist = float("inf")
4     pheromones = calcPheromones(places)
5     visibility = calcVisibility(matrix, places)
6     ants = places
7     for day in range(days):
8         route = np.arange(places)
9         visited = calcVisitedPlaces(route, ants)
10        for ant in range(ants):
11            while (len(visited[ant]) != ants):
12                pk = findWays(pheromones, visibility, visited,
13                             places, ant, alpha, beta)
14                chosenPlace = chooseNextPlaceByPosibility(pk)
15                visited[ant].append(chosenPlace - 1)
16
17            visited[ant].append(visited[ant][0])
18
19            curLength = calcLength(matrix, visited[ant])
20
21            if (curLength < minDist):
22                minDist = curLength
23                bestWay = visited[ant]
24
25        pheromones = updatePheromones(matrix, places, visited,
26                                       pheromones, q, k_evaporation)
```

Листинг 3.3 – Реализация алгоритма нахождения массива вероятностей переходов в непосещенные города

```
1 def findWays(pheromones, visibility, visited, places, ant,
2   alpha, beta):
3
4     for place in range(places):
5         if place not in visited[ant]:
6             ant_place = visited[ant][-1]
7             pk[place] = pow(pheromones[ant_place][place],
8                             alpha) * \
9                             pow(visibility[ant_place][place], beta)
10        else:
11            pk[place] = 0
12
13    sum_pk = sum(pk)
14
15    for place in range(places):
16        pk[place] /= sum_pk
17
18    return pk
```

Листинг 3.4 – Реализация алгоритма нахождения массива вероятностей переходов в непосещенные города

```
1 def calcPheromones(size):
2     min_phero = 1
3     pheromones = [[min_phero for i in range(size)] for j in
4                    range(size)]
5     return pheromones
```

Листинг 3.5 – Реализация алгоритма выбора следующего города

```
1 def chooseNextPlaceByPosibility(pk):
2     posibility = random()
3     choice = 0
4     chosenPlace = 0
5     while ((choice < posibility) and (chosenPlace < len(pk))):
6         choice += pk[chosenPlace]
7         chosenPlace += 1
8
9     return chosenPlace
```

Листинг 3.6 – Реализация алгоритма обновления матрицы феромонов

```
1 def updatePheromones(matrix, places, visited, pheromones, q,  
    k_evaporation):  
2     ants = places  
3  
4     for i in range(places):  
5         for j in range(places):  
6             delta = 0  
7             for ant in range(ants):  
8                 length = calcLength(matrix, visited[ant])  
9                 delta += q / length  
10  
11             pheromones[i][j] *= (1 - k_evaporation)  
12             pheromones[i][j] += delta  
13             if (pheromones[i][j] < MIN_PHEROMONE):  
14                 pheromones[i][j] = MIN_PHEROMONE  
15  
16     return pheromones
```

3.4 Функциональные тесты

В таблице 3.1 приведены тесты для функций программы. Все функциональные тесты пройдены *успешно*.

Таблица 3.1 – Функциональные тесты

Матрица смежности	Ожидаемый результат	Результат программы
$\begin{pmatrix} 0 & 4 & 2 & 1 & 7 \\ 4 & 0 & 3 & 7 & 2 \\ 2 & 3 & 0 & 10 & 3 \\ 1 & 7 & 10 & 0 & 9 \\ 7 & 2 & 3 & 9 & 0 \end{pmatrix}$	15, [0, 2, 4, 1, 3, 0]	15, [0, 2, 4, 1, 3, 0]
$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$	4, [0, 1, 2, 0]	4, [0, 1, 2, 0]
$\begin{pmatrix} 0 & 15 & 19 & 20 \\ 15 & 0 & 12 & 13 \\ 19 & 12 & 0 & 17 \\ 20 & 13 & 17 & 0 \end{pmatrix}$	64, [0, 1, 2, 3, 0]	64, [0, 1, 2, 3, 0]

Вывод

Были представлены листинги всех реализаций алгоритмов — полного перебора и муравьиного. Также в данном разделе была приведена информация о выбранных средствах для разработки алгоритмов и сведения о модулях программы, проведено функциональное тестирование.

4 Исследовательская часть

В данном разделе будет приведен пример работы программы, а также проведен сравнительный анализ алгоритмов при различных ситуациях на основе полученных данных.

4.1 Технические характеристики

Технические характеристики устройства, на котором выполнялись замеры по времени представлены далее.

- Процессор: Intel(R) Core(TM) i5-10300H CPU 2.50 ГГц [5].
- Количество ядер: 4 физических и 8 логических ядер.
- Оперативная память: 16 ГБайт.
- Операционная система: Windows 11 Pro 64-разрядная система [6].

При замерах времени ноутбук был включен в сеть электропитания и был нагружен только системными приложениями.

4.2 Демонстрация работы программы

На рисунке 4.1 представлен пример работы программы для обоих алгоритмов — полного перебора и муравьиного. Осуществляется выбор файла с данными, ввод коэффициентов для муравьиного алгоритма, а также выполнение алгоритма решения задачи коммивояжера методом полного перебора и методом на основе муравьиного алгоритма.

```
Меню
1. Полный перебор
2. Муравьиный алгоритм
3. Все алгоритмы
4. Параметризация
5. Замерить время
6. Обновить данные
7. Распечатать матрицу
0. Выход
Выбор: 3

Введите коэффициент alpha: 0.7
Введите коэффициент evaporation: 0.8
Введите кол-во дней: 10

Алгоритм полного перебора
Минимальная длина пути = 80
Путь: [0, 1, 2, 3, 4, 5, 6, 7, 0]

Муравьиный алгоритм
Минимальная длина пути = 80
Путь: [0, 7, 6, 5, 4, 3, 2, 1, 0]
```

Рисунок 4.1 – Пример работы программы

4.3 Временные характеристики

Для замера процессорного времени используется функция `process_time(...)` из библиотеки `time` на *Python*. Функция возвращает процессорное время типа `float` в секундах.

Использовать функцию приходится дважды, затем из конечного времени нужно вычесть начальное, чтобы получить результат.

Замеры проводились для разного размера матриц, чтобы определить, когда наиболее эффективно использовать муравьиный алгоритм.

Результаты замеров приведены в таблице 4.1 (время в с.).

Таблица 4.1 – Результаты замеров времени (в с.)

Кол-во заявок	Время, мкс	
	Полный перебор	Полный перебор
2	0.000128	0.018732
3	0.000138	0.029615
4	0.000244	0.040381
5	0.000478	0.079291
6	0.02509	0.122331
7	0.019703	0.254059
8	0.16285	0.398472
9	1.777611	0.494024
10	9.227853	0.957666

На рисунке 4.2 приведен график результатов замеров времени работы реализаций алгоритмов для различных линейных размеров матриц.

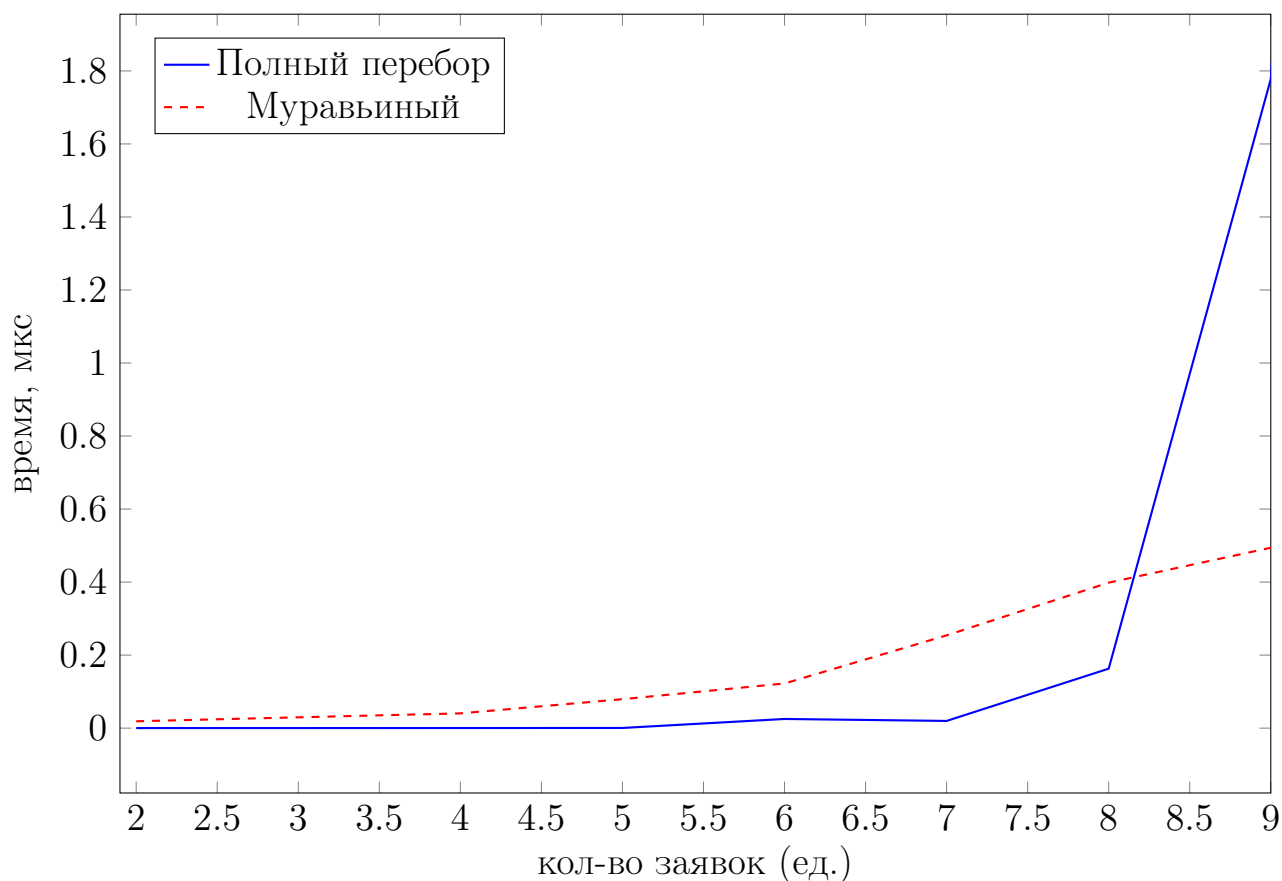


Рисунок 4.2 – Результаты замеров времени работы реализации конвейерной обработки

4.4 Постановка эксперимента

Автоматическая параметризация была проведена на двух классах данных — 4.4.1 и 4.4.2. Алгоритм будет запущен для набора значений $\alpha, \rho \in (0, 1)$.

Итоговая таблица значений параметризации будет состоять из следующих колонок:

- α — коэффициент жадности;
- ρ — коэффициент испарения;
- *days* — количество дней жизни колонии муравьев;
- *Result* — эталонный результат, полученный методом полного перебора для проведения данного эксперимента;
- *Mistake* — разность полученного основанного на муравьином алгоритме метода значения и эталонного значения на данных значениях параметров, показатель качества решения.

Цель эксперимента — определить комбинацию параметров, которые позволяют решить задачу наилучшим образом для выбранного класса данных. Качество решения зависит от количества дней и погрешности измерений.

4.4.1 Класс данных 1

Согласно с вариантом представим граф, вершины которого будут являться городами путешествия за 80 дней:

- 1) Лондон;
- 2) Суэц;
- 3) Бомбей;
- 4) Калькутта;

- 5) Гонконг;
- 6) Йокогама;
- 7) Сан-Франциско;
- 8) Нью-Йорк.

Занимаемый путь из:

- 1) Лондон в Суэц за 7 дней;
- 2) Суэц в Бомбей за 13 дней;
- 3) Бомбей в Калькутта за 3 дня;
- 4) Калькутта в Гонконг за 13 дней;
- 5) Гонконг в Йокогама за 6 дней;
- 6) Йокогама в Сан-Франциско за 22 дня;
- 7) Сан-Франциско в Нью-Йорк за 7 дней;
- 8) Нью-Йорк в Лондон за 9 дней.

Класс данных 1 представляет собой матрицу смежности размером 8 элементов (небольшой разброс значений — от 1 до 50), которая представлена далее.

$$K_1 = \begin{pmatrix} 0 & 7 & 21 & 24 & 37 & 39 & 7 & 9 \\ 7 & 0 & 13 & 17 & 30 & 41 & 19 & 12 \\ 21 & 13 & 0 & 3 & 17 & 23 & 12 & 32 \\ 24 & 17 & 3 & 0 & 13 & 27 & 35 & 43 \\ 37 & 30 & 17 & 13 & 0 & 6 & 27 & 36 \\ 39 & 41 & 23 & 27 & 6 & 0 & 22 & 30 \\ 17 & 19 & 29 & 35 & 27 & 22 & 0 & 7 \\ 9 & 12 & 32 & 43 & 36 & 30 & 7 & 0 \end{pmatrix} \quad (4.1)$$

Для данного класса данных приведена таблица 4.2 – 4.4 с выборкой параметров, которые наилучшим образом решают поставленную задачу,

полные результаты параметризации приведены в приложении А. Используются следующие обозначения: Days — количество дней, Result — результат работы, Mistake — ошибка как отклонение решения от эталонного.

В выборке, разделенной на подгруппы по признаку значения параметра α , для пары (α, ρ) выбран набор значений параметров, обеспечивающих наилучший результат приближения (наименьшее значение параметра mistake). Если одинаковый результат параметра mistake достигается для нескольких кортежей $(\alpha, \rho, days, result, mistake)$, содержащих одинаковые значения параметров (α, ρ) , среди них выбирается кортеж, содержащий наименьшее значение параметра days.

Таблица 4.2 – Выборка из параметров для класса данных 1 (Начало)

α	ρ	Days	Result	Mistake
0.1	0.1	1	80	0
0.1	0.2	5	80	0
0.1	0.3	5	80	0
0.1	0.4	10	80	0
0.1	0.5	5	80	0
0.1	0.6	50	80	0
0.1	0.7	50	80	0
0.1	0.8	5	80	0
0.2	0.1	10	80	0
0.2	0.2	50	80	0
0.2	0.3	10	80	0
0.2	0.4	10	80	0
0.2	0.5	50	80	0
0.2	0.6	50	80	0
0.2	0.7	10	80	0
0.2	0.8	50	80	0
0.3	0.1	10	80	0
0.3	0.2	100	80	0
0.3	0.3	50	80	0
0.3	0.4	10	80	0

Таблица 4.3 – Выборка из параметров для класса данных 1
(Продолжение)

α	ρ	Days	Result	Mistake
0.3	0.5	100	80	0
0.3	0.6	50	80	0
0.3	0.7	100	80	0
0.3	0.8	50	80	0
0.4	0.1	50	80	0
0.4	0.2	50	80	0
0.4	0.3	50	80	0
0.4	0.4	10	80	0
0.4	0.5	50	80	0
0.4	0.6	50	80	0
0.4	0.7	50	80	0
0.4	0.8	50	80	0
0.5	0.1	50	80	0
0.5	0.2	50	80	0
0.5	0.3	50	80	0
0.5	0.4	10	80	0
0.5	0.5	50	80	0
0.5	0.6	100	80	0
0.5	0.7	50	80	0
0.5	0.8	300	80	0
0.6	0.1	100	80	0
0.6	0.2	50	80	0
0.6	0.3	100	80	0
0.6	0.4	100	80	0
0.6	0.5	50	80	0
0.6	0.6	50	80	0
0.6	0.7	10	80	0
0.6	0.8	100	80	0
0.7	0.1	100	80	0
0.7	0.2	50	80	0

Таблица 4.4 – Выборка из параметров для класса данных 1
(Продолжение)

α	ρ	Days	Result	Mistake
0.7	0.3	50	80	0
0.7	0.4	100	80	0
0.7	0.5	50	80	0
0.7	0.6	50	80	0
0.7	0.7	10	80	0
0.8	0.8	10	80	0
0.8	0.1	50	80	0
0.8	0.2	300	80	0
0.8	0.3	100	80	0
0.8	0.4	300	80	0
0.8	0.5	300	80	0
0.8	0.6	100	80	0
0.8	0.7	300	80	0
0.8	0.8	50	80	0
0.9	0.1	300	80	0
0.9	0.2	300	80	0
0.9	0.3	100	80	0
0.9	0.4	80	80	0
0.9	0.5	300	80	0
0.9	0.6	50	80	0
0.9	0.7	100	80	0
0.9	0.8	50	80	0

4.4.2 Класс данных 2

Класс данных 2 представляет собой матрицу смежности размером 9 элементов (большой разброс значений - от 1000 до 9999), которая представлена далее.

$$K_1 = \begin{pmatrix} 0 & 1470 & 6489 & 2010 & 9573 & 2842 & 4881 & 5450 \\ 1470 & 0 & 3721 & 9794 & 4278 & 6202 & 3552 & 1825 \\ 1825 & 3721 & 0 & 6856 & 6856 & 8202 & 7770 & 9909 \\ 4924 & 9794 & 6856 & 0 & 4036 & 3150 & 8496 & 5701 \\ 9573 & 4278 & 1177 & 4036 & 0 & 4467 & 3438 & 4887 \\ 2842 & 6202 & 8202 & 3150 & 4467 & 0 & 6736 & 3139 \\ 4881 & 3552 & 7770 & 8496 & 3438 & 6736 & 0 & 3716 \\ 5450 & 1825 & 9909 & 5701 & 4887 & 3139 & 3716 & 0 \end{pmatrix} \quad (4.2)$$

Для данного класса данных приведена таблица 4.5 – 4.7 с выборкой параметров, которые наилучшим образом решают поставленную задачу, полные результаты параметризации приведены в приложении Б.

Таблица 4.5 – Выборка из параметров для класса данных 2 (Начало)

α	ρ	Days	Result	Mistake
0.1	0.1	50	24474	0
0.1	0.2	3	24474	0
0.1	0.3	3	24474	0
0.1	0.4	1	24474	0
0.1	0.5	10	24474	0
0.1	0.6	50	24474	0
0.1	0.7	100	24474	0
0.1	0.8	50	24474	0
0.2	0.1	10	24474	0
0.2	0.2	5	24474	0
0.2	0.3	50	24474	0
0.2	0.4	10	24474	0
0.2	0.5	3	24474	0
0.2	0.6	50	24474	0
0.2	0.7	50	24474	0
0.2	0.8	10	24474	0
0.3	0.1	10	24474	0
0.3	0.2	100	24474	0
0.3	0.3	5	24474	0
0.3	0.4	10	24474	0
0.3	0.5	100	24474	0
0.3	0.6	50	24474	0
0.3	0.7	100	24474	0
0.3	0.8	50	24474	0
0.4	0.1	50	24474	0
0.4	0.2	100	24474	0
0.4	0.3	50	24474	0
0.4	0.4	50	24474	0
0.4	0.5	100	24474	0

Таблица 4.6 – Выборка из параметров для класса данных 2
(Продолжение)

α	ρ	Days	Result	Mistake
0.4	0.6	50	24474	0
0.4	0.7	10	24474	0
0.4	0.8	50	24474	0
0.5	0.1	300	24474	0
0.5	0.2	100	24474	0
0.5	0.3	50	24474	0
0.5	0.4	300	24474	0
0.5	0.5	300	24474	0
0.5	0.6	300	24474	0
0.5	0.7	50	24474	0
0.5	0.8	300	24474	0
0.6	0.1	50	24474	0
0.6	0.2	100	24474	0
0.6	0.3	100	24474	0
0.6	0.4	300	24474	0
0.6	0.5	50	24474	0
0.6	0.6	300	24474	0
0.6	0.7	10	24474	0
0.6	0.8	50	24474	0
0.7	0.1	100	24474	0
0.7	0.2	100	24474	0
0.7	0.3	50	24474	0
0.7	0.4	50	24474	0
0.7	0.5	50	24474	0
0.7	0.6	100	24474	0
0.7	0.7	300	24474	0
0.8	0.8	300	24474	0
0.8	0.1	50	24474	0
0.8	0.2	50	24474	0

Таблица 4.7 – Выборка из параметров для класса данных 2
(Продолжение)

α	ρ	Days	Result	Mistake
0.8	0.3	100	24474	0
0.8	0.4	100	24474	0
0.8	0.5	10	24474	0
0.8	0.6	50	24474	0
0.8	0.7	300	24474	0
0.8	0.8	50	24474	0
0.9	0.1	100	24474	0
0.9	0.2	300	24474	0
0.9	0.3	300	24474	0
0.9	0.4	100	24474	0
0.9	0.5	100	24474	0
0.9	0.6	100	24474	0
0.9	0.7	100	24474	0
0.9	0.8	300	24474	0

4.5 Вывод

В результате эксперимента было получено, что использование муравьиного алгоритма наиболее эффективно при больших размерах матриц. Так, при размере матрицы, равном 2, муравьиный алгоритм менее алгоритма полного перебора в 143 раза, а при размере матрицы, равном 9, муравьиный алгоритм быстрее алгоритма полного перебора в раз, а при размере в 10 – уже в 15 раз. Следовательно, при размерах матриц больше 8 следует использовать муравьиный алгоритм, но стоит учитывать, что он не гарантирует получения глобального оптимума при решении задачи.

Также при проведении эксперимента с классами данных было получено, что на первом классе данных (см. п. 4.4.1) муравьиный алгоритм лучше всего показывает себя при параметрах:

- $\alpha = 0.1, \rho = 0.1, days = 1;$
- $\alpha = 0.2, \rho \in \{0.1, 0.3, 0.4, 0.7\}, days = 10;$
- $\alpha = 0.3, \rho \in \{0.1, 0.4\}, days = 10;$
- $\alpha = 0.4, \rho = 0.4, days = 10;$
- $\alpha = 0.5, \rho = 0.4, days = 10;$
- $\alpha = 0.6, \rho = 0.7, days = 10;$
- $\alpha = 0.7, \rho \in \{0.7, 0.8\}, days = 10.$

Следовательно, для класса данных 1 рекомендуется использовать данные параметры.

Для класса данных 2 (см. п. 4.4.2) было получено, что наилучшим образом алгоритм работает на значениях параметров, которые представлены далее:

- $\alpha = 0.1, \rho = 0.4, days = 1;$
- $\alpha = 0.2, \rho = 0.5, days = 3;$
- $\alpha = 0.3, \rho = 0.3, days = 5;$
- $\alpha = 0.4, \rho = 0.7, days = 10;$
- $\alpha = 0.6, \rho = 0.7, days = 10;$
- $\alpha = 0.8, \rho = 0.5, days = 10.$

Для второго класса данных 2 рекомендуется использовать данные параметры.

Заключение

Поставленная цель достигнута: получен навык параметризация методов на примере решения задачи коммивояжера методом на основе муравьиного алгоритма.

В ходе выполнения лабораторной работы были решены все задачи:

- 1) Описана задача коммивояжера.
- 2) Описаны методы решения задачи коммивояжера — метод полного перебора и метод на основе муравьиного алгоритма.
- 3) Приведены схемы муравьиного алгоритма и алгоритма, позволяющего решить задачу коммивояжера методом полного перебора.
- 4) Разработан и реализован программный продукт, позволяющий решить задачу коммивояжера исследуемыми методами.
- 5) Сравнены по времени метод полного перебора и метод на основе муравьиного алгоритма.
- 6) Описаны и обоснованы полученные результаты в отчете о выполненной лабораторной работе.

Исходя из полученных результатов, использование муравьиного алгоритма наиболее эффективно по времени при больших размерах матриц. Так при размере матрицы, равном 2, муравьиный алгоритм медленнее алгоритма полного перебора в 143 раза, а при размере матрицы, равном 9, муравьиный алгоритм быстрее алгоритма полного перебора в раз, а при размере в 10 – уже в 15 раз. Следовательно, при размерах матриц больше 8 следует использовать муравьиный алгоритм, но стоит учитывать, что он не гарантирует оптимального решения, в отличие от метода полного перебора.

Список использованных источников

- 1 О. Борознов В. Исследование решения задачи коммивояжера. — АГТУ, Вестник Астраханского государственного технического университета. [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/issledovanie-resheniya-zadachi-kommivoyazhera/viewer> (дата обращения: 23.01.2023).
- 2 Семёнов С. С. Педан А. В. Воловиков В. С. Климов И. С. Анализ трудоёмкости различных алгоритмических подходов для решения задачи коммивояжера — ООО «Корпорация «Интел Групп» [Электронный ресурс]. Режим доступа: [https://cyberleninka.ru/article/n/analiz-trudoemkosti-razlichnyh-algoritmicheskikh-podhodov-dlya-\resheniya-zadachi-kommivoyazhera](https://cyberleninka.ru/article/n/analiz-trudoemkosti-razlichnyh-algoritmicheskikh-podhodov-dlya-resheniya-zadachi-kommivoyazhera) (дата обращения: 23.01.2023).
- 3 Welcome to Python [Электронный ресурс]. Режим доступа: <https://www.python.org> (дата обращения: 23.01.2023).
- 4 time — Time access and conversions [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/library/time.html#functions> (дата обращения: 23.01.2023).
- 5 Intel [Электронный ресурс]. — Режим доступа: <https://ark.intel.com/content/www/ru/ru/ark/products/201839/intel-core-i510300h-processor-8m-cache-up-to-4-50-ghz.html> (дата обращения: 25.09.2022).
- 6 Windows 10 Pro 2h21 64-bit [Электронный ресурс]. — Режим доступа: <https://www.microsoft.com/ru-ru/software-download/windows10> (дата обращения: 25.09.2022).

Приложение А

Таблица 4.8 – Параметризация для класса данных 1, Days — количество дней, Result — результат работы, Mistake — ошибочность полученного результата

α	ρ	Days	Result	Mistake
0.1	0.1	3	80	8
0.1	0.1	5	80	19
0.1	0.1	10	80	0
0.1	0.1	50	80	0
0.1	0.1	100	80	0
0.1	0.1	300	80	0
0.1	0.1	500	80	0
0.1	0.2	1	80	60
0.1	0.2	3	80	8
0.1	0.2	5	80	0
0.1	0.2	10	80	0
0.1	0.2	50	80	0
0.1	0.2	100	80	0
0.1	0.2	300	80	0
0.1	0.2	500	80	0
0.1	0.3	1	80	26
0.1	0.3	3	80	26
0.1	0.3	5	80	0
0.1	0.3	10	80	0
0.1	0.3	50	80	0
0.1	0.3	100	80	0
0.1	0.3	300	80	0
0.1	0.3	500	80	0
0.1	0.4	1	80	46
0.1	0.4	3	80	8
0.1	0.4	5	80	8
0.1	0.4	10	80	0

0.1	0.4	50	80	0
0.1	0.4	100	80	0
0.1	0.4	300	80	0
0.1	0.4	500	80	0
0.1	0.5	1	80	19
0.1	0.5	3	80	34
0.1	0.5	5	80	0
0.1	0.5	10	80	0
0.1	0.5	50	80	0
0.1	0.5	100	80	0
0.1	0.5	300	80	0
0.1	0.5	500	80	0
0.1	0.6	1	80	35
0.1	0.6	3	80	19
0.1	0.6	5	80	11
0.1	0.6	10	80	11
0.1	0.6	50	80	0
0.1	0.6	100	80	0
0.1	0.6	300	80	0
0.1	0.6	500	80	0
0.1	0.7	1	80	29
0.1	0.7	3	80	0
0.1	0.7	5	80	30
0.1	0.7	10	80	8
0.1	0.7	50	80	0
0.1	0.7	100	80	0
0.1	0.7	300	80	0
0.1	0.7	500	80	0
0.1	0.8	1	80	36
0.1	0.8	3	80	11
0.1	0.8	5	80	0
0.1	0.8	10	80	0
0.1	0.8	50	80	0

0.1	0.8	100	80	0
0.1	0.8	300	80	0
0.1	0.8	500	80	0
0.2	0.1	1	80	24
0.2	0.1	3	80	31
0.2	0.1	5	80	11
0.2	0.1	10	80	0
0.2	0.1	50	80	0
0.2	0.1	100	80	0
0.2	0.1	300	80	0
0.2	0.1	500	80	0
0.2	0.2	1	80	16
0.2	0.2	3	80	19
0.2	0.2	5	80	11
0.2	0.2	10	80	16
0.2	0.2	50	80	0
0.2	0.2	100	80	0
0.2	0.2	300	80	0
0.2	0.2	500	80	0
0.2	0.3	1	80	33
0.2	0.3	3	80	29
0.2	0.3	5	80	19
0.2	0.3	10	80	0
0.2	0.3	50	80	0
0.2	0.3	100	80	0
0.2	0.3	300	80	0
0.2	0.3	500	80	0
0.2	0.4	1	80	8
0.2	0.4	3	80	8
0.2	0.4	5	80	18
0.2	0.4	10	80	0
0.2	0.4	50	80	0
0.2	0.4	100	80	0

0.2	0.4	300	80	0
0.2	0.4	500	80	0
0.2	0.5	1	80	34
0.2	0.5	3	80	31
0.2	0.5	5	80	0
0.2	0.5	10	80	18
0.2	0.5	50	80	0
0.2	0.5	100	80	0
0.2	0.5	300	80	0
0.2	0.5	500	80	0
0.2	0.6	1	80	38
0.2	0.6	3	80	30
0.2	0.6	5	80	16
0.2	0.6	10	80	8
0.2	0.6	50	80	0
0.2	0.6	100	80	0
0.2	0.6	300	80	0
0.2	0.6	500	80	0
0.2	0.7	1	80	23
0.2	0.7	3	80	15
0.2	0.7	5	80	11
0.2	0.7	10	80	0
0.2	0.7	50	80	0
0.2	0.7	100	80	0
0.2	0.7	300	80	0
0.2	0.7	500	80	0
0.2	0.8	1	80	32
0.2	0.8	3	80	29
0.2	0.8	5	80	0
0.2	0.8	10	80	15
0.2	0.8	50	80	0
0.2	0.8	100	80	0
0.2	0.8	300	80	0

0.2	0.8	500	80	0
0.3	0.1	1	80	38
0.3	0.1	3	80	33
0.3	0.1	5	80	0
0.3	0.1	10	80	8
0.3	0.1	50	80	0
0.3	0.1	100	80	0
0.3	0.1	300	80	0
0.3	0.1	500	80	0
0.3	0.2	1	80	34
0.3	0.2	3	80	11
0.3	0.2	5	80	29
0.3	0.2	10	80	18
0.3	0.2	50	80	0
0.3	0.2	100	80	0
0.3	0.2	300	80	0
0.3	0.2	500	80	0
0.3	0.3	1	80	41
0.3	0.3	3	80	18
0.3	0.3	5	80	11
0.3	0.3	10	80	18
0.3	0.3	50	80	0
0.3	0.3	100	80	0
0.3	0.3	300	80	0
0.3	0.3	500	80	0
0.3	0.4	1	80	24
0.3	0.4	3	80	16
0.3	0.4	5	80	38
0.3	0.4	10	80	8
0.3	0.4	50	80	0
0.3	0.4	100	80	0
0.3	0.4	300	80	0
0.3	0.4	500	80	0

0.3	0.5	1	80	25
0.3	0.5	3	80	26
0.3	0.5	5	80	11
0.3	0.5	10	80	8
0.3	0.5	50	80	0
0.3	0.5	100	80	0
0.3	0.5	300	80	0
0.3	0.5	500	80	0
0.3	0.6	1	80	31
0.3	0.6	3	80	18
0.3	0.6	5	80	18
0.3	0.6	10	80	15
0.3	0.6	50	80	0
0.3	0.6	100	80	0
0.3	0.6	300	80	0
0.3	0.6	500	80	0
0.3	0.7	1	80	23
0.3	0.7	3	80	30
0.3	0.7	5	80	19
0.3	0.7	10	80	11
0.3	0.7	50	80	0
0.3	0.7	100	80	0
0.3	0.7	300	80	0
0.3	0.7	500	80	0
0.3	0.8	1	80	0
0.3	0.8	3	80	8
0.3	0.8	5	80	19
0.3	0.8	10	80	11
0.3	0.8	50	80	0
0.3	0.8	100	80	0
0.3	0.8	300	80	0
0.3	0.8	500	80	0
0.4	0.1	1	80	18

0.4	0.1	3	80	29
0.4	0.1	5	80	8
0.4	0.1	10	80	23
0.4	0.1	50	80	0
0.4	0.1	100	80	0
0.4	0.1	300	80	0
0.4	0.1	500	80	0
0.4	0.2	1	80	50
0.4	0.2	3	80	19
0.4	0.2	5	80	16
0.4	0.2	10	80	19
0.4	0.2	50	80	0
0.4	0.2	100	80	0
0.4	0.2	300	80	0
0.4	0.2	500	80	0
0.4	0.3	1	80	30
0.4	0.3	3	80	29
0.4	0.3	5	80	23
0.4	0.3	10	80	19
0.4	0.3	50	80	0
0.4	0.3	100	80	0
0.4	0.3	300	80	0
0.4	0.3	500	80	0
0.4	0.4	1	80	19
0.4	0.4	3	80	41
0.4	0.4	5	80	32
0.4	0.4	10	80	0
0.4	0.4	50	80	0
0.4	0.4	100	80	0
0.4	0.4	300	80	0
0.4	0.4	500	80	0
0.4	0.5	1	80	42
0.4	0.5	3	80	8

0.4	0.5	5	80	16
0.4	0.5	10	80	8
0.4	0.5	50	80	0
0.4	0.5	100	80	0
0.4	0.5	300	80	0
0.4	0.5	500	80	0
0.4	0.6	1	80	34
0.4	0.6	3	80	18
0.4	0.6	5	80	19
0.4	0.6	10	80	8
0.4	0.6	50	80	0
0.4	0.6	100	80	0
0.4	0.6	300	80	0
0.4	0.6	500	80	0
0.4	0.7	1	80	54
0.4	0.7	3	80	34
0.4	0.7	5	80	18
0.4	0.7	10	80	11
0.4	0.7	50	80	0
0.4	0.7	100	80	0
0.4	0.7	300	80	0
0.4	0.7	500	80	0
0.4	0.8	1	80	24
0.4	0.8	3	80	11
0.4	0.8	5	80	11
0.4	0.8	10	80	8
0.4	0.8	50	80	0
0.4	0.8	100	80	0
0.4	0.8	300	80	0
0.4	0.8	500	80	0
0.5	0.1	1	80	58
0.5	0.1	3	80	39
0.5	0.1	5	80	19

0.5	0.1	10	80	18
0.5	0.1	50	80	0
0.5	0.1	100	80	0
0.5	0.1	300	80	0
0.5	0.1	500	80	0
0.5	0.2	1	80	16
0.5	0.2	3	80	35
0.5	0.2	5	80	35
0.5	0.2	10	80	29
0.5	0.2	50	80	0
0.5	0.2	100	80	0
0.5	0.2	300	80	0
0.5	0.2	500	80	0
0.5	0.3	1	80	11
0.5	0.3	3	80	23
0.5	0.3	5	80	0
0.5	0.3	10	80	25
0.5	0.3	50	80	0
0.5	0.3	100	80	0
0.5	0.3	300	80	0
0.5	0.3	500	80	0
0.5	0.4	1	80	46
0.5	0.4	3	80	23
0.5	0.4	5	80	8
0.5	0.4	10	80	0
0.5	0.4	50	80	0
0.5	0.4	100	80	0
0.5	0.4	300	80	0
0.5	0.4	500	80	0
0.5	0.5	1	80	35
0.5	0.5	3	80	37
0.5	0.5	5	80	15
0.5	0.5	10	80	19

0.5	0.5	50	80	0
0.5	0.5	100	80	0
0.5	0.5	300	80	0
0.5	0.5	500	80	0
0.5	0.6	1	80	33
0.5	0.6	3	80	31
0.5	0.6	5	80	8
0.5	0.6	10	80	33
0.5	0.6	50	80	8
0.5	0.6	100	80	0
0.5	0.6	300	80	0
0.5	0.6	500	80	0
0.5	0.7	1	80	47
0.5	0.7	3	80	30
0.5	0.7	5	80	30
0.5	0.7	10	80	0
0.5	0.7	50	80	0
0.5	0.7	100	80	0
0.5	0.7	300	80	0
0.5	0.7	500	80	0
0.5	0.8	1	80	38
0.5	0.8	3	80	25
0.5	0.8	5	80	19
0.5	0.8	10	80	8
0.5	0.8	50	80	8
0.5	0.8	100	80	8
0.5	0.8	300	80	0
0.5	0.8	500	80	0
0.6	0.1	1	80	50
0.6	0.1	3	80	34
0.6	0.1	5	80	30
0.6	0.1	10	80	11
0.6	0.1	50	80	18

0.6	0.1	100	80	0
0.6	0.1	300	80	0
0.6	0.1	500	80	0
0.6	0.2	1	80	35
0.6	0.2	3	80	11
0.6	0.2	5	80	0
0.6	0.2	10	80	19
0.6	0.2	50	80	0
0.6	0.2	100	80	0
0.6	0.2	300	80	0
0.6	0.2	500	80	0
0.6	0.3	1	80	51
0.6	0.3	3	80	37
0.6	0.3	5	80	30
0.6	0.3	10	80	0
0.6	0.3	50	80	11
0.6	0.3	100	80	0
0.6	0.3	300	80	0
0.6	0.3	500	80	0
0.6	0.4	1	80	43
0.6	0.4	3	80	36
0.6	0.4	5	80	15
0.6	0.4	10	80	30
0.6	0.4	50	80	11
0.6	0.4	100	80	0
0.6	0.4	300	80	0
0.6	0.4	500	80	0
0.6	0.5	1	80	42
0.6	0.5	3	80	29
0.6	0.5	5	80	8
0.6	0.5	10	80	24
0.6	0.5	50	80	0
0.6	0.5	100	80	0

0.6	0.5	300	80	0
0.6	0.5	500	80	0
0.6	0.6	1	80	28
0.6	0.6	3	80	31
0.6	0.6	5	80	40
0.6	0.6	10	80	15
0.6	0.6	50	80	0
0.6	0.6	100	80	0
0.6	0.6	300	80	0
0.6	0.6	500	80	0
0.6	0.7	1	80	23
0.6	0.7	3	80	46
0.6	0.7	5	80	33
0.6	0.7	10	80	0
0.6	0.7	50	80	0
0.6	0.7	100	80	0
0.6	0.7	300	80	0
0.6	0.7	500	80	0
0.6	0.8	1	80	58
0.6	0.8	3	80	29
0.6	0.8	5	80	31
0.6	0.8	10	80	28
0.6	0.8	50	80	8
0.6	0.8	100	80	0
0.6	0.8	300	80	0
0.6	0.8	500	80	0
0.7	0.1	1	80	51
0.7	0.1	3	80	36
0.7	0.1	5	80	24
0.7	0.1	10	80	0
0.7	0.1	50	80	19
0.7	0.1	100	80	0
0.7	0.1	300	80	0

0.7	0.1	500	80	0
0.7	0.2	1	80	38
0.7	0.2	3	80	36
0.7	0.2	5	80	36
0.7	0.2	10	80	35
0.7	0.2	50	80	0
0.7	0.2	100	80	0
0.7	0.2	300	80	0
0.7	0.2	500	80	0
0.7	0.3	1	80	0
0.7	0.3	3	80	48
0.7	0.3	5	80	33
0.7	0.3	10	80	36
0.7	0.3	50	80	0
0.7	0.3	100	80	18
0.7	0.3	300	80	0
0.7	0.3	500	80	0
0.7	0.4	1	80	24
0.7	0.4	3	80	30
0.7	0.4	5	80	19
0.7	0.4	10	80	16
0.7	0.4	50	80	11
0.7	0.4	100	80	0
0.7	0.4	300	80	0
0.7	0.4	500	80	0
0.7	0.5	1	80	52
0.7	0.5	3	80	31
0.7	0.5	5	80	19
0.7	0.5	10	80	35
0.7	0.5	50	80	0
0.7	0.5	100	80	0
0.7	0.5	300	80	0
0.7	0.5	500	80	0

0.7	0.6	1	80	36
0.7	0.6	3	80	34
0.7	0.6	5	80	15
0.7	0.6	10	80	0
0.7	0.6	50	80	0
0.7	0.6	100	80	0
0.7	0.6	300	80	0
0.7	0.6	500	80	0
0.7	0.7	1	80	51
0.7	0.7	3	80	49
0.7	0.7	5	80	31
0.7	0.7	10	80	0
0.7	0.7	50	80	11
0.7	0.7	100	80	0
0.7	0.7	300	80	0
0.7	0.7	500	80	0
0.7	0.8	1	80	53
0.7	0.8	3	80	35
0.7	0.8	5	80	28
0.7	0.8	10	80	0
0.7	0.8	50	80	11
0.7	0.8	100	80	0
0.7	0.8	300	80	0
0.7	0.8	500	80	0
0.8	0.1	1	80	35
0.8	0.1	3	80	11
0.8	0.1	5	80	35
0.8	0.1	10	80	31
0.8	0.1	50	80	8
0.8	0.1	100	80	8
0.8	0.1	300	80	0
0.8	0.1	500	80	0
0.8	0.2	1	80	58

0.8	0.2	3	80	48
0.8	0.2	5	80	16
0.8	0.2	10	80	19
0.8	0.2	50	80	0
0.8	0.2	100	80	8
0.8	0.2	300	80	0
0.8	0.2	500	80	0
0.8	0.3	1	80	58
0.8	0.3	3	80	19
0.8	0.3	5	80	36
0.8	0.3	10	80	33
0.8	0.3	50	80	15
0.8	0.3	100	80	0
0.8	0.3	300	80	0
0.8	0.3	500	80	0
0.8	0.4	1	80	63
0.8	0.4	3	80	41
0.8	0.4	5	80	35
0.8	0.4	10	80	18
0.8	0.4	50	80	18
0.8	0.4	100	80	18
0.8	0.4	300	80	0
0.8	0.4	500	80	0
0.8	0.5	1	80	18
0.8	0.5	3	80	34
0.8	0.5	5	80	16
0.8	0.5	10	80	34
0.8	0.5	50	80	16
0.8	0.5	100	80	0
0.8	0.5	300	80	0
0.8	0.5	500	80	0
0.8	0.6	1	80	68
0.8	0.6	3	80	42

0.8	0.6	5	80	35
0.8	0.6	10	80	30
0.8	0.6	50	80	15
0.8	0.6	100	80	0
0.8	0.6	300	80	0
0.8	0.6	500	80	0
0.8	0.7	1	80	42
0.8	0.7	3	80	33
0.8	0.7	5	80	23
0.8	0.7	10	80	31
0.8	0.7	50	80	19
0.8	0.7	100	80	19
0.8	0.7	300	80	0
0.8	0.7	500	80	0
0.8	0.8	1	80	59
0.8	0.8	3	80	34
0.8	0.8	5	80	19
0.8	0.8	10	80	44
0.8	0.8	50	80	0
0.8	0.8	100	80	0
0.8	0.8	300	80	0
0.8	0.8	500	80	0
0.9	0.1	1	80	65
0.9	0.1	3	80	29
0.9	0.1	5	80	35
0.9	0.1	10	80	39
0.9	0.1	50	80	16
0.9	0.1	100	80	18
0.9	0.1	300	80	0
0.9	0.1	500	80	0
0.9	0.2	1	80	30
0.9	0.2	3	80	60
0.9	0.2	5	80	48

0.9	0.2	10	80	42
0.9	0.2	50	80	8
0.9	0.2	100	80	8
0.9	0.2	300	80	0
0.9	0.2	500	80	0
0.9	0.3	1	80	75
0.9	0.3	3	80	30
0.9	0.3	5	80	43
0.9	0.3	10	80	18
0.9	0.3	50	80	11
0.9	0.3	100	80	0
0.9	0.3	300	80	0
0.9	0.3	500	80	0
0.9	0.4	1	80	36
0.9	0.4	3	80	0
0.9	0.4	5	80	46
0.9	0.4	10	80	29
0.9	0.4	50	80	18
0.9	0.4	100	80	8
0.9	0.4	300	80	0
0.9	0.4	500	80	0
0.9	0.5	1	80	34
0.9	0.5	3	80	35
0.9	0.5	5	80	42
0.9	0.5	10	80	36
0.9	0.5	50	80	8
0.9	0.5	100	80	8
0.9	0.5	300	80	0
0.9	0.5	500	80	0
0.9	0.6	1	80	63
0.9	0.6	3	80	38
0.9	0.6	5	80	44
0.9	0.6	10	80	44

0.9	0.6	50	80	0
0.9	0.6	100	80	0
0.9	0.6	300	80	8
0.9	0.6	500	80	0
0.9	0.7	1	80	26
0.9	0.7	3	80	56
0.9	0.7	5	80	40
0.9	0.7	10	80	32
0.9	0.7	50	80	16
0.9	0.7	100	80	0
0.9	0.7	300	80	0
0.9	0.7	500	80	0
0.9	0.8	1	80	44
0.9	0.8	3	80	16
0.9	0.8	5	80	40
0.9	0.8	10	80	36
0.9	0.8	50	80	0
0.9	0.8	100	80	8
0.9	0.8	300	80	0
0.9	0.8	500	80	0

Приложение Б

Таблица 4.9 – Параметризация для класса данных 2, Days — количество дней, Result — результат работы, Mistake — ошибочность полученного результата

α	ρ	Days	Result	Mistake
0.1	0.1	3	24474	261
0.1	0.1	5	24474	2255
0.1	0.1	10	24474	261
0.1	0.1	50	24474	0
0.1	0.1	100	24474	0
0.1	0.1	300	24474	0
0.1	0.1	500	24474	0
0.1	0.2	1	24474	4985
0.1	0.2	3	24474	0
0.1	0.2	5	24474	816
0.1	0.2	10	24474	261
0.1	0.2	50	24474	0
0.1	0.2	100	24474	0
0.1	0.2	300	24474	0
0.1	0.2	500	24474	0
0.1	0.3	1	24474	6670
0.1	0.3	3	24474	0
0.1	0.3	5	24474	0
0.1	0.3	10	24474	0
0.1	0.3	50	24474	0
0.1	0.3	100	24474	0
0.1	0.3	300	24474	0
0.1	0.3	500	24474	0
0.1	0.4	1	24474	0
0.1	0.4	3	24474	816
0.1	0.4	5	24474	4827
0.1	0.4	10	24474	0

0.1	0.4	50	24474	0
0.1	0.4	100	24474	0
0.1	0.4	300	24474	0
0.1	0.4	500	24474	0
0.1	0.5	1	24474	0
0.1	0.5	3	24474	1512
0.1	0.5	5	24474	4925
0.1	0.5	10	24474	0
0.1	0.5	50	24474	0
0.1	0.5	100	24474	0
0.1	0.5	300	24474	0
0.1	0.5	500	24474	0
0.1	0.6	1	24474	6508
0.1	0.6	3	24474	874
0.1	0.6	5	24474	4193
0.1	0.6	10	24474	816
0.1	0.6	50	24474	0
0.1	0.6	100	24474	261
0.1	0.6	300	24474	0
0.1	0.6	500	24474	0
0.1	0.7	1	24474	1781
0.1	0.7	3	24474	816
0.1	0.7	5	24474	4696
0.1	0.7	10	24474	0
0.1	0.7	50	24474	741
0.1	0.7	100	24474	0
0.1	0.7	300	24474	0
0.1	0.7	500	24474	0
0.1	0.8	1	24474	1512
0.1	0.8	3	24474	3279
0.1	0.8	5	24474	3170
0.1	0.8	10	24474	2255
0.1	0.8	50	24474	0

0.1	0.8	100	24474	0
0.1	0.8	300	24474	0
0.1	0.8	500	24474	0
0.2	0.1	1	24474	0
0.2	0.1	3	24474	5248
0.2	0.1	5	24474	5510
0.2	0.1	10	24474	0
0.2	0.1	50	24474	261
0.2	0.1	100	24474	0
0.2	0.1	300	24474	0
0.2	0.1	500	24474	0
0.2	0.2	1	24474	8101
0.2	0.2	3	24474	1512
0.2	0.2	5	24474	0
0.2	0.2	10	24474	0
0.2	0.2	50	24474	0
0.2	0.2	100	24474	0
0.2	0.2	300	24474	0
0.2	0.2	500	24474	0
0.2	0.3	1	24474	2633
0.2	0.3	3	24474	816
0.2	0.3	5	24474	3170
0.2	0.3	10	24474	2261
0.2	0.3	50	24474	0
0.2	0.3	100	24474	0
0.2	0.3	300	24474	0
0.2	0.3	500	24474	0
0.2	0.4	1	24474	8155
0.2	0.4	3	24474	0
0.2	0.4	5	24474	1512
0.2	0.4	10	24474	0
0.2	0.4	50	24474	0
0.2	0.4	100	24474	0

0.2	0.4	300	24474	0
0.2	0.4	500	24474	0
0.2	0.5	1	24474	2255
0.2	0.5	3	24474	741
0.2	0.5	5	24474	3220
0.2	0.5	10	24474	0
0.2	0.5	50	24474	0
0.2	0.5	100	24474	0
0.2	0.5	300	24474	0
0.2	0.5	500	24474	0
0.2	0.6	1	24474	8794
0.2	0.6	3	24474	0
0.2	0.6	5	24474	3428
0.2	0.6	10	24474	1462
0.2	0.6	50	24474	0
0.2	0.6	100	24474	0
0.2	0.6	300	24474	0
0.2	0.6	500	24474	0
0.2	0.7	1	24474	6843
0.2	0.7	3	24474	4827
0.2	0.7	5	24474	741
0.2	0.7	10	24474	741
0.2	0.7	50	24474	0
0.2	0.7	100	24474	0
0.2	0.7	300	24474	0
0.2	0.7	500	24474	0
0.2	0.8	1	24474	5150
0.2	0.8	3	24474	1781
0.2	0.8	5	24474	5307
0.2	0.8	10	24474	0
0.2	0.8	50	24474	0
0.2	0.8	100	24474	0
0.2	0.8	300	24474	0

0.2	0.8	500	24474	0
0.3	0.1	1	24474	741
0.3	0.1	3	24474	3428
0.3	0.1	5	24474	4251
0.3	0.1	10	24474	0
0.3	0.1	50	24474	0
0.3	0.1	100	24474	0
0.3	0.1	300	24474	0
0.3	0.1	500	24474	0
0.3	0.2	1	24474	5832
0.3	0.2	3	24474	6347
0.3	0.2	5	24474	5595
0.3	0.2	10	24474	741
0.3	0.2	50	24474	261
0.3	0.2	100	24474	0
0.3	0.2	300	24474	0
0.3	0.2	500	24474	0
0.3	0.3	1	24474	6375
0.3	0.3	3	24474	3428
0.3	0.3	5	24474	0
0.3	0.3	10	24474	1462
0.3	0.3	50	24474	0
0.3	0.3	100	24474	0
0.3	0.3	300	24474	0
0.3	0.3	500	24474	0
0.3	0.4	1	24474	7870
0.3	0.4	3	24474	4193
0.3	0.4	5	24474	874
0.3	0.4	10	24474	0
0.3	0.4	50	24474	741
0.3	0.4	100	24474	0
0.3	0.4	300	24474	0
0.3	0.4	500	24474	0

0.3	0.5	1	24474	7880
0.3	0.5	3	24474	261
0.3	0.5	5	24474	5088
0.3	0.5	10	24474	816
0.3	0.5	50	24474	816
0.3	0.5	100	24474	0
0.3	0.5	300	24474	0
0.3	0.5	500	24474	0
0.3	0.6	1	24474	6205
0.3	0.6	3	24474	4982
0.3	0.6	5	24474	2255
0.3	0.6	10	24474	261
0.3	0.6	50	24474	0
0.3	0.6	100	24474	0
0.3	0.6	300	24474	0
0.3	0.6	500	24474	0
0.3	0.7	1	24474	3220
0.3	0.7	3	24474	3220
0.3	0.7	5	24474	741
0.3	0.7	10	24474	1462
0.3	0.7	50	24474	261
0.3	0.7	100	24474	0
0.3	0.7	300	24474	0
0.3	0.7	500	24474	0
0.3	0.8	1	24474	5512
0.3	0.8	3	24474	874
0.3	0.8	5	24474	5248
0.3	0.8	10	24474	816
0.3	0.8	50	24474	0
0.3	0.8	100	24474	0
0.3	0.8	300	24474	0
0.3	0.8	500	24474	0
0.4	0.1	1	24474	3758

0.4	0.1	3	24474	3428
0.4	0.1	5	24474	3775
0.4	0.1	10	24474	2974
0.4	0.1	50	24474	0
0.4	0.1	100	24474	0
0.4	0.1	300	24474	0
0.4	0.1	500	24474	0
0.4	0.2	1	24474	7315
0.4	0.2	3	24474	5567
0.4	0.2	5	24474	261
0.4	0.2	10	24474	741
0.4	0.2	50	24474	741
0.4	0.2	100	24474	0
0.4	0.2	300	24474	0
0.4	0.2	500	24474	0
0.4	0.3	1	24474	7083
0.4	0.3	3	24474	3701
0.4	0.3	5	24474	3279
0.4	0.3	10	24474	741
0.4	0.3	50	24474	0
0.4	0.3	100	24474	0
0.4	0.3	300	24474	0
0.4	0.3	500	24474	0
0.4	0.4	1	24474	7088
0.4	0.4	3	24474	3701
0.4	0.4	5	24474	0
0.4	0.4	10	24474	261
0.4	0.4	50	24474	0
0.4	0.4	100	24474	261
0.4	0.4	300	24474	0
0.4	0.4	500	24474	0
0.4	0.5	1	24474	5270
0.4	0.5	3	24474	5890

0.4	0.5	5	24474	3701
0.4	0.5	10	24474	0
0.4	0.5	50	24474	741
0.4	0.5	100	24474	0
0.4	0.5	300	24474	0
0.4	0.5	500	24474	0
0.4	0.6	1	24474	3428
0.4	0.6	3	24474	4125
0.4	0.6	5	24474	3220
0.4	0.6	10	24474	2255
0.4	0.6	50	24474	0
0.4	0.6	100	24474	0
0.4	0.6	300	24474	0
0.4	0.6	500	24474	0
0.4	0.7	1	24474	6898
0.4	0.7	3	24474	4985
0.4	0.7	5	24474	4982
0.4	0.7	10	24474	0
0.4	0.7	50	24474	0
0.4	0.7	100	24474	0
0.4	0.7	300	24474	0
0.4	0.7	500	24474	0
0.4	0.8	1	24474	3701
0.4	0.8	3	24474	5630
0.4	0.8	5	24474	2974
0.4	0.8	10	24474	816
0.4	0.8	50	24474	0
0.4	0.8	100	24474	261
0.4	0.8	300	24474	0
0.4	0.8	500	24474	0
0.5	0.1	1	24474	11188
0.5	0.1	3	24474	3700
0.5	0.1	5	24474	1462

0.5	0.1	10	24474	261
0.5	0.1	50	24474	816
0.5	0.1	100	24474	261
0.5	0.1	300	24474	0
0.5	0.1	500	24474	0
0.5	0.2	1	24474	4710
0.5	0.2	3	24474	1462
0.5	0.2	5	24474	1781
0.5	0.2	10	24474	3279
0.5	0.2	50	24474	816
0.5	0.2	100	24474	0
0.5	0.2	300	24474	0
0.5	0.2	500	24474	0
0.5	0.3	1	24474	5091
0.5	0.3	3	24474	874
0.5	0.3	5	24474	1781
0.5	0.3	10	24474	816
0.5	0.3	50	24474	0
0.5	0.3	100	24474	0
0.5	0.3	300	24474	0
0.5	0.3	500	24474	0
0.5	0.4	1	24474	2633
0.5	0.4	3	24474	2261
0.5	0.4	5	24474	3775
0.5	0.4	10	24474	2974
0.5	0.4	50	24474	0
0.5	0.4	100	24474	261
0.5	0.4	300	24474	0
0.5	0.4	500	24474	0
0.5	0.5	1	24474	5088
0.5	0.5	3	24474	6603
0.5	0.5	5	24474	4193
0.5	0.5	10	24474	1462

0.5	0.5	50	24474	741
0.5	0.5	100	24474	261
0.5	0.5	300	24474	0
0.5	0.5	500	24474	0
0.5	0.6	1	24474	6765
0.5	0.6	3	24474	3700
0.5	0.6	5	24474	2633
0.5	0.6	10	24474	741
0.5	0.6	50	24474	261
0.5	0.6	100	24474	261
0.5	0.6	300	24474	0
0.5	0.6	500	24474	0
0.5	0.7	1	24474	5328
0.5	0.7	3	24474	2974
0.5	0.7	5	24474	3387
0.5	0.7	10	24474	1462
0.5	0.7	50	24474	0
0.5	0.7	100	24474	0
0.5	0.7	300	24474	0
0.5	0.7	500	24474	0
0.5	0.8	1	24474	7945
0.5	0.8	3	24474	6604
0.5	0.8	5	24474	2255
0.5	0.8	10	24474	3701
0.5	0.8	50	24474	816
0.5	0.8	100	24474	261
0.5	0.8	300	24474	0
0.5	0.8	500	24474	0
0.6	0.1	1	24474	10314
0.6	0.1	3	24474	9298
0.6	0.1	5	24474	816
0.6	0.1	10	24474	2974
0.6	0.1	50	24474	0

0.6	0.1	100	24474	0
0.6	0.1	300	24474	0
0.6	0.1	500	24474	0
0.6	0.2	1	24474	6387
0.6	0.2	3	24474	5270
0.6	0.2	5	24474	1462
0.6	0.2	10	24474	1781
0.6	0.2	50	24474	261
0.6	0.2	100	24474	0
0.6	0.2	300	24474	0
0.6	0.2	500	24474	0
0.6	0.3	1	24474	4575
0.6	0.3	3	24474	7187
0.6	0.3	5	24474	874
0.6	0.3	10	24474	261
0.6	0.3	50	24474	741
0.6	0.3	100	24474	0
0.6	0.3	300	24474	0
0.6	0.3	500	24474	0
0.6	0.4	1	24474	9595
0.6	0.4	3	24474	4744
0.6	0.4	5	24474	3809
0.6	0.4	10	24474	261
0.6	0.4	50	24474	741
0.6	0.4	100	24474	816
0.6	0.4	300	24474	0
0.6	0.4	500	24474	0
0.6	0.5	1	24474	9373
0.6	0.5	3	24474	5091
0.6	0.5	5	24474	261
0.6	0.5	10	24474	261
0.6	0.5	50	24474	0
0.6	0.5	100	24474	741

0.6	0.5	300	24474	0
0.6	0.5	500	24474	0
0.6	0.6	1	24474	874
0.6	0.6	3	24474	5512
0.6	0.6	5	24474	3428
0.6	0.6	10	24474	816
0.6	0.6	50	24474	874
0.6	0.6	100	24474	261
0.6	0.6	300	24474	0
0.6	0.6	500	24474	0
0.6	0.7	1	24474	9582
0.6	0.7	3	24474	261
0.6	0.7	5	24474	4575
0.6	0.7	10	24474	0
0.6	0.7	50	24474	816
0.6	0.7	100	24474	0
0.6	0.7	300	24474	0
0.6	0.7	500	24474	0
0.6	0.8	1	24474	10709
0.6	0.8	3	24474	4575
0.6	0.8	5	24474	4503
0.6	0.8	10	24474	2633
0.6	0.8	50	24474	0
0.6	0.8	100	24474	0
0.6	0.8	300	24474	0
0.6	0.8	500	24474	0
0.7	0.1	1	24474	7945
0.7	0.1	3	24474	5890
0.7	0.1	5	24474	5890
0.7	0.1	10	24474	3387
0.7	0.1	50	24474	741
0.7	0.1	100	24474	0
0.7	0.1	300	24474	0

0.7	0.1	500	24474	0
0.7	0.2	1	24474	3775
0.7	0.2	3	24474	1781
0.7	0.2	5	24474	3220
0.7	0.2	10	24474	2974
0.7	0.2	50	24474	1462
0.7	0.2	100	24474	0
0.7	0.2	300	24474	0
0.7	0.2	500	24474	0
0.7	0.3	1	24474	8478
0.7	0.3	3	24474	3700
0.7	0.3	5	24474	3428
0.7	0.3	10	24474	741
0.7	0.3	50	24474	0
0.7	0.3	100	24474	0
0.7	0.3	300	24474	0
0.7	0.3	500	24474	261
0.7	0.4	1	24474	9632
0.7	0.4	3	24474	741
0.7	0.4	5	24474	7826
0.7	0.4	10	24474	816
0.7	0.4	50	24474	0
0.7	0.4	100	24474	0
0.7	0.4	300	24474	0
0.7	0.4	500	24474	0
0.7	0.5	1	24474	5832
0.7	0.5	3	24474	3387
0.7	0.5	5	24474	741
0.7	0.5	10	24474	4827
0.7	0.5	50	24474	0
0.7	0.5	100	24474	0
0.7	0.5	300	24474	0
0.7	0.5	500	24474	0

0.7	0.6	1	24474	8636
0.7	0.6	3	24474	7129
0.7	0.6	5	24474	4251
0.7	0.6	10	24474	2261
0.7	0.6	50	24474	261
0.7	0.6	100	24474	0
0.7	0.6	300	24474	0
0.7	0.6	500	24474	0
0.7	0.7	1	24474	4696
0.7	0.7	3	24474	3758
0.7	0.7	5	24474	3816
0.7	0.7	10	24474	3176
0.7	0.7	50	24474	261
0.7	0.7	100	24474	261
0.7	0.7	300	24474	0
0.7	0.7	500	24474	0
0.7	0.8	1	24474	9478
0.7	0.8	3	24474	1512
0.7	0.8	5	24474	5307
0.7	0.8	10	24474	1781
0.7	0.8	50	24474	0
0.7	0.8	100	24474	1462
0.7	0.8	300	24474	0
0.7	0.8	500	24474	0
0.8	0.1	1	24474	3701
0.8	0.1	3	24474	3428
0.8	0.1	5	24474	3809
0.8	0.1	10	24474	3170
0.8	0.1	50	24474	0
0.8	0.1	100	24474	261
0.8	0.1	300	24474	0
0.8	0.1	500	24474	0
0.8	0.2	1	24474	6765

0.8	0.2	3	24474	7679
0.8	0.2	5	24474	3279
0.8	0.2	10	24474	1462
0.8	0.2	50	24474	0
0.8	0.2	100	24474	816
0.8	0.2	300	24474	0
0.8	0.2	500	24474	0
0.8	0.3	1	24474	7304
0.8	0.3	3	24474	6547
0.8	0.3	5	24474	6502
0.8	0.3	10	24474	5370
0.8	0.3	50	24474	261
0.8	0.3	100	24474	261
0.8	0.3	300	24474	0
0.8	0.3	500	24474	0
0.8	0.4	1	24474	5370
0.8	0.4	3	24474	4251
0.8	0.4	5	24474	5307
0.8	0.4	10	24474	4632
0.8	0.4	50	24474	741
0.8	0.4	100	24474	0
0.8	0.4	300	24474	741
0.8	0.4	500	24474	0
0.8	0.5	1	24474	6547
0.8	0.5	3	24474	6447
0.8	0.5	5	24474	6604
0.8	0.5	10	24474	0
0.8	0.5	50	24474	741
0.8	0.5	100	24474	261
0.8	0.5	300	24474	0
0.8	0.5	500	24474	0
0.8	0.6	1	24474	5595
0.8	0.6	3	24474	10078

0.8	0.6	5	24474	741
0.8	0.6	10	24474	4957
0.8	0.6	50	24474	0
0.8	0.6	100	24474	261
0.8	0.6	300	24474	0
0.8	0.6	500	24474	0
0.8	0.7	1	24474	6765
0.8	0.7	3	24474	5270
0.8	0.7	5	24474	2633
0.8	0.7	10	24474	2974
0.8	0.7	50	24474	1462
0.8	0.7	100	24474	741
0.8	0.7	300	24474	0
0.8	0.7	500	24474	0
0.8	0.8	1	24474	6843
0.8	0.8	3	24474	5248
0.8	0.8	5	24474	5643
0.8	0.8	10	24474	5567
0.8	0.8	50	24474	0
0.8	0.8	100	24474	0
0.8	0.8	300	24474	0
0.8	0.8	500	24474	0
0.9	0.1	1	24474	4710
0.9	0.1	3	24474	10321
0.9	0.1	5	24474	3700
0.9	0.1	10	24474	2633
0.9	0.1	50	24474	741
0.9	0.1	100	24474	0
0.9	0.1	300	24474	0
0.9	0.1	500	24474	0
0.9	0.2	1	24474	4710
0.9	0.2	3	24474	6508
0.9	0.2	5	24474	6387

0.9	0.2	10	24474	1512
0.9	0.2	50	24474	261
0.9	0.2	100	24474	2255
0.9	0.2	300	24474	0
0.9	0.2	500	24474	0
0.9	0.3	1	24474	11153
0.9	0.3	3	24474	2261
0.9	0.3	5	24474	4503
0.9	0.3	10	24474	2255
0.9	0.3	50	24474	1512
0.9	0.3	100	24474	874
0.9	0.3	300	24474	0
0.9	0.3	500	24474	0
0.9	0.4	1	24474	8501
0.9	0.4	3	24474	7244
0.9	0.4	5	24474	4125
0.9	0.4	10	24474	1781
0.9	0.4	50	24474	261
0.9	0.4	100	24474	0
0.9	0.4	300	24474	741
0.9	0.4	500	24474	0
0.9	0.5	1	24474	6697
0.9	0.5	3	24474	8277
0.9	0.5	5	24474	1512
0.9	0.5	10	24474	1781
0.9	0.5	50	24474	2261
0.9	0.5	100	24474	0
0.9	0.5	300	24474	0
0.9	0.5	500	24474	0
0.9	0.6	1	24474	4982
0.9	0.6	3	24474	4985
0.9	0.6	5	24474	1781
0.9	0.6	10	24474	3428

0.9	0.6	50	24474	874
0.9	0.6	100	24474	0
0.9	0.6	300	24474	0
0.9	0.6	500	24474	0
0.9	0.7	1	24474	13715
0.9	0.7	3	24474	8448
0.9	0.7	5	24474	2633
0.9	0.7	10	24474	741
0.9	0.7	50	24474	2261
0.9	0.7	100	24474	0
0.9	0.7	300	24474	0
0.9	0.7	500	24474	0
0.9	0.8	1	24474	9198
0.9	0.8	3	24474	6347
0.9	0.8	5	24474	7315
0.9	0.8	10	24474	4696
0.9	0.8	50	24474	816
0.9	0.8	100	24474	1512
0.9	0.8	300	24474	0
0.9	0.8	500	24474	0