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"

Тема Муравьиный алгоритм

Студент Цветков И.А.

Группа ИУ7-53Б

Оценка (баллы)

Преподаватель Волкова Л. Л.

Москва — 2021 г.

Содержание

Введение **3**

1 Аналитическая часть **4**

1.1 Задача коммивояжера 4

1.2 Алгоритм полного перебора для решения задачи коммивояжера 4

1.3 Муравьиный алгоритм 4

1.4 Вывод 6

2 Конструкторская часть **7**

2.1 Описание используемых типов данных 7

2.2 Структура разрабатываемого ПО 7

2.3 Схемы алгоритмов 8

2.4 Классы эквивалентности при тестировании 15

2.5 Вывод 15

3 Технологическая часть **16**

3.1 Средства реализации 16

3.2 Листинги кода 16

3.3 Сведения о модулях программы 20

3.4 Функциональные тесты 21

3.5 Вывод 21

4 Исследовательская часть **22**

4.1 Технические характеристики 22

4.2 Демонстрация работы программы 22

4.3 Время выполнения алгоритмов 24

4.4 Постановка эксперимента 26

4.4.1 Класс данных 1 26

4.4.2 Класс данных 2 28

4.5 Вывод 30

Заключение **32**

Список литературы **33**

Приложение 1 **34**

Приложение 2 **52**

Введение

Во все времена важной задачей была оптимизация, которая позволяла ускорять работу существующих алгоритмов или предлагать совершенно новые пути решения для увеличения скорости выполнения той или иной задачи. Одной из важных задач является логистическая задача поисков оптимальных маршрутов. Чаще всего они решаются полным перебором, что крайне неэффективно при большом количестве вершин в графе (каждую задачу поиска оптимального маршрута можно представить в виде графа – набора вершин и ребер).

Целью данной работы является изучение задачи коммивояжера, которая решается муравьиным алгоритмом. Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить задачу коммивояжера;
- описать алгоритмы решения задачи коммивояжера – полный перебор и муравьиный алгоритм;
- привести схемы полного перебора и муравьиного алгоритмов;
- описать используемые типы и структуры данных;
- описать структуру разрабатываемого программного обеспечения;
- реализовать разработанные алгоритмы;
- провести функциональное тестирование разработанного алгоритма;
- провести сравнительный анализ по времени для реализованного алгоритма;
- подготовить отчет по лабораторной работе.

1 Аналитическая часть

В этом разделе будет представлена информация о задаче коммивояжера, а также о путях ее решения – полным перебором или муравьиным алгоритмом.

1.1 Задача коммивояжера

Задача коммивояжера [1] – (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). Суть – найти такой порядок посещения вершин графа, при котором путь будет минимален, каждая вершина будет посещена лишь один раз, а возврат произойдет в начальную вершину.

1.2 Алгоритм полного перебора для решения задачи коммивояжера

Полный перебор для задачи коммивояжера [2] – имеет высокую сложность алгоритма ($n!$). 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.

1.3 Муравьиный алгоритм

Муравьиный алгоритм [3] – основан на принципе поведения колонии муравьев.

Муравьи действуют, ощущая некий феромон. Каждый муравей, чтобы другие могли ориентироваться, оставляет на своем пути феромоны. При большом количестве прохождения муравьев, наибольшее количество феромона остается на оптимальном пути.

Суть в том, что отдельно взятый муравей мало, что может, поскольку способен выполнять только максимально простые задачи. Но при условии большого количества таких муравьев они могу самоорганизовываться в большие очереди для решения сложных задач.

Пусть муравей имеет следующие характеристики:

- зрение – способен определить длину ребра;
- память – запоминает пройденный маршрут;
- обоняние – чувствует феромон.

Также введем целевую функцию (1.1).

$$\eta_{ij} = 1/D_{ij}, \tag{1.1}$$

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

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

$$P_{kij} = \begin{cases} \frac{\tau_{ij}^a \eta_{ij}^b}{\sum_{q=1}^{m} \tau_{iq}^a \eta_{iq}^b}, \text{вершина не была посещена ранее муравьем k}, \\ 0, \text{иначе} \end{cases} \tag{1.2}$$

где a – параметр влияния длины пути, b – параметр влияния феромона, τ_{ij} – расстояния от города i до j, η_{ij} – количество феромонов на ребре ij.

После завершения движения всех муравьев, формула обновляется феромон по формуле (1.3):

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

При этом

$$\Delta\tau_{ij} = \sum_{k=1}^{N} \tau_{ij}^k, \tag{1.4}$$

где

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, \text{ребро посещено k-ым муравьем}, \\ 0, \text{иначе} \end{cases} \tag{1.5}$$

Путь выбирается по следующей схеме:

1. Каждый муравей имеет список запретов – список уже посещенных городов (вершин графа).

2. Муравьиной зрение – отвечает за желание посетить вершину.

3. Муравьиное обоняние – отвечает за ощущение феромона на определенном пути (ребре). При этом количество феромона на пути (ребре) в момент времени t обозначается как $\tau_{i,j}(t)$.

4. После прохождения определенного ребра муравей откладывает на нем некотрое количество феромона, которое показывает оптимальность сделанного выбора (это кол-во вычисляется по формуле (1.5))

1.4 Вывод

В данном разделе была рассмотрена задача коммивояжера, а также полный перебор для ее решения и муравьиный алгоритм.

Программа будет получать на вход матрицу смежности, для которой можно будет выбрать один из алгоритмов поиска путей – полным перебором или муравьиный. Также можно будет ввести коэффиценты и количество дней для работы муравьиного алгоритма. При неверном вводе какого-то из значений будет выдаваться сообщение об ошибке.

Реализуемое ПО дает возможность получить минимальную сумму пути, а также сам путь, используя один из алгоритмов. Также имеется возможность провести тестирование по времени для разных размеров матриц.

2 Конструкторская часть

В этом разделе будут представлено описание используемых типов данных, а также схемы алгоритма полного перебора и муравьиного алгоритма.

2.1 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

- размер матрицы смежности - целое число типа *int*;
- название файла - строка типа *str*;
- коэффициент α, β, *evaporation_koef* - числа типа *float*;
- матрица смежностей - матрица типа *int* для хранения длины путей между городами.

2.2 Структура разрабатываемого ПО

В данном ПО будет реализован метод структурного программирования. Для взаимодействия с пользователем будет разработано меню, которое будет предоставлять возможность выбрать нужный алгоритм - полного перебора или муравьиный, замерить время, провести параметризацию муравьиного алгоритма, а также построить графики для сравнения времени выполнения алгоритмов.

Для работы будут разработаны следующие процедуры:

- процедура, реализующая генерацию квадратной матрицы значениями в определенном диапазоне по ее размеру, входные данные - размер матрицы, диапазон значений (start_num, end_num), выходные - полученная матрица;
- процедура вывода полученной матрицы на экран (для отладки), входные данные - матрица, выходные - выведенная на экран матрица;

- процедура, реализующая алгоритм полного перебора путей, входные данные - матрица, количество городов (размер матрицы), выходные - длина найденного минимального пути и сам путь;

- процедура, реализующая муравьиный алгоритм перебора путей, входные данные - матрица, количество городов (размер матрицы), коэффициенты α, β, $evaporation\_koef$, кол-во дней, выходные - длина найденного минимального пути и сам путь;

- процедуры замера времени алгоритма полного перебора и муравьиного алгоритма на различных размерах матриц, входные данные - начальный размер матрицы, конечный размер матрицы, выходные - результаты замеров времени;

- процедура для построения графиков по полученным временным замерам, входные данные - замеры времени, выходные - график.

2.3 Схемы алгоритмов

На рисунке 2.1 представлена схема алгоритма полного перебора путей, а на рисунках 2.2–2.3 схема муравьиного алгоритма поиска путей. Также на рисунках 2.4–2.6 представлены схемы вспомогательных функций для муравьиного алгоритма.

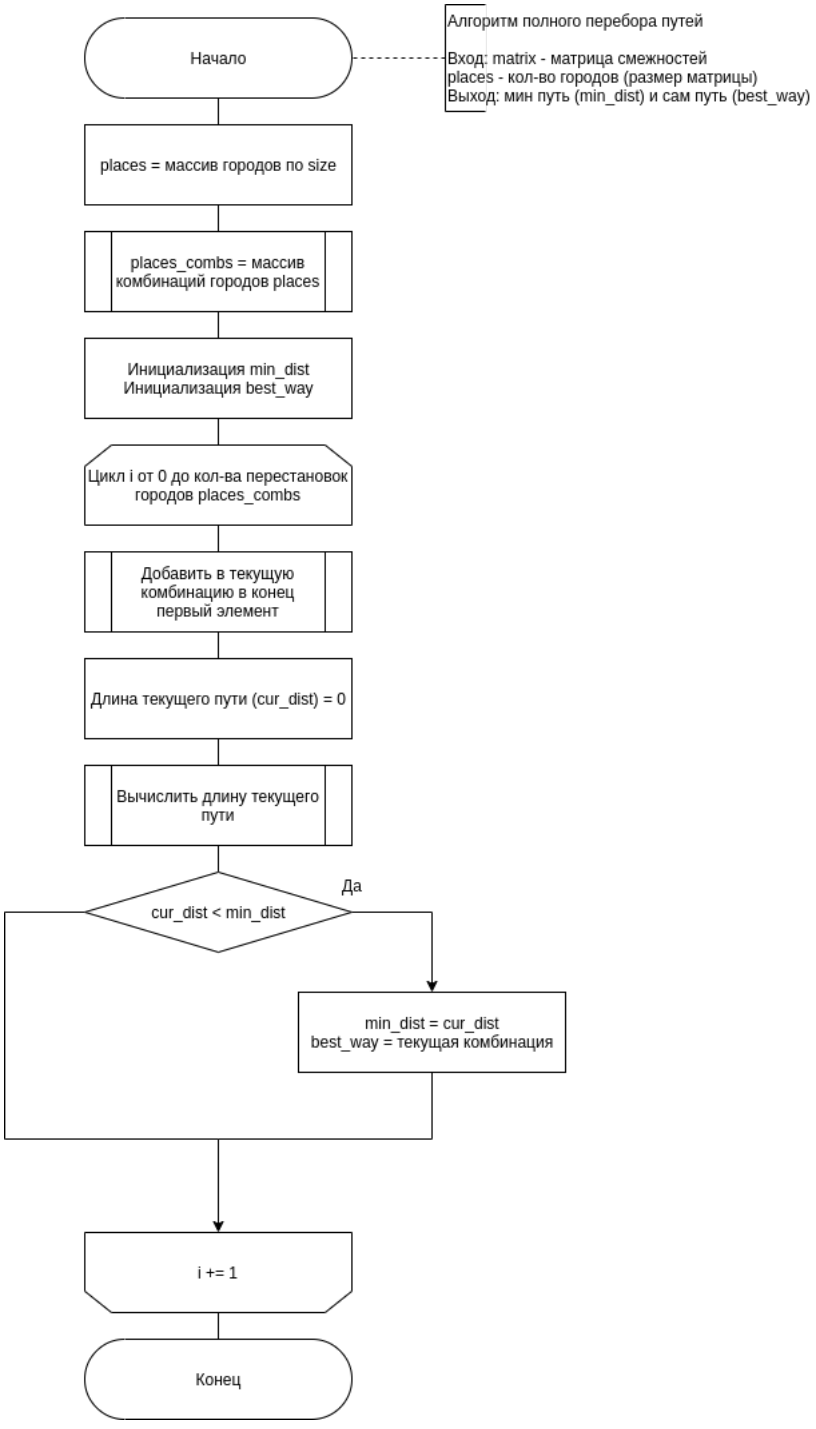

Рисунок 2.1 – Схема алгоритма полного перебора путей

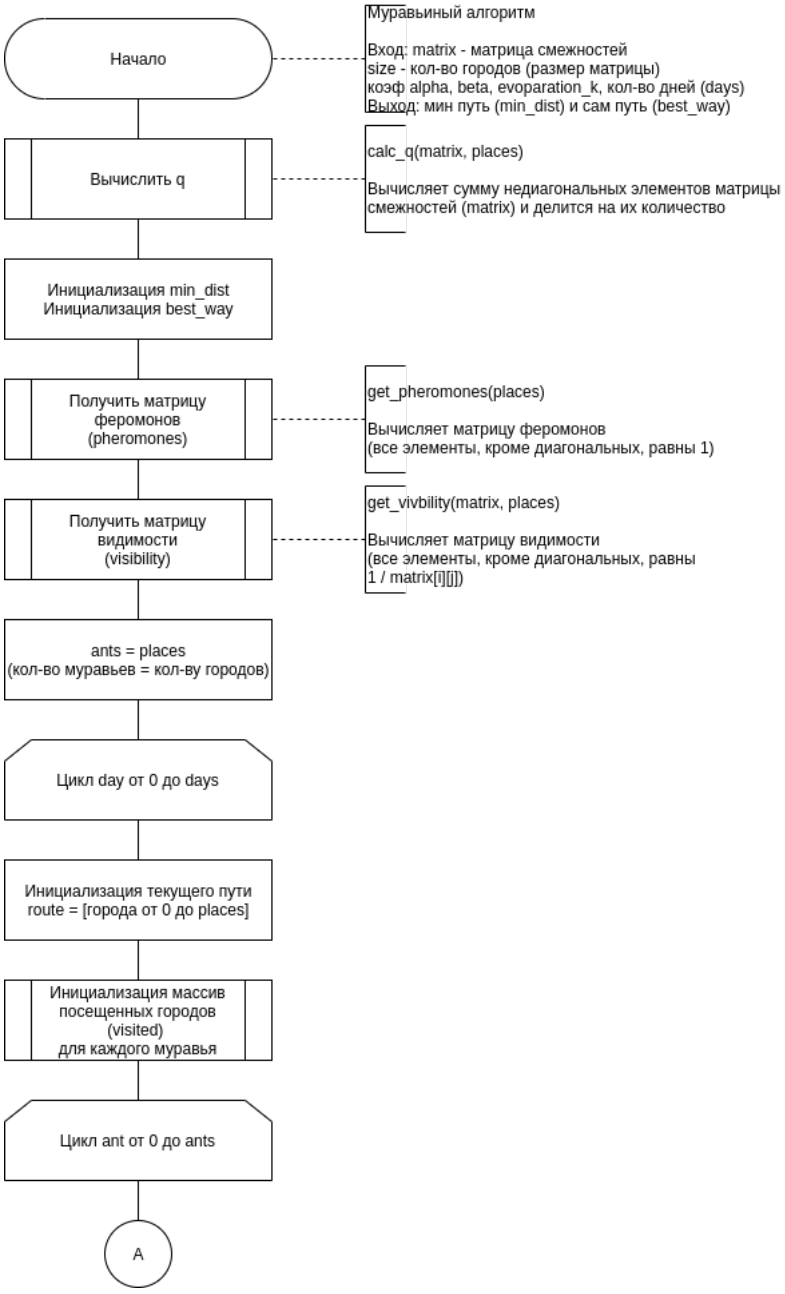

Рисунок 2.2 – Схема муравьиного алгоритма (часть 1)

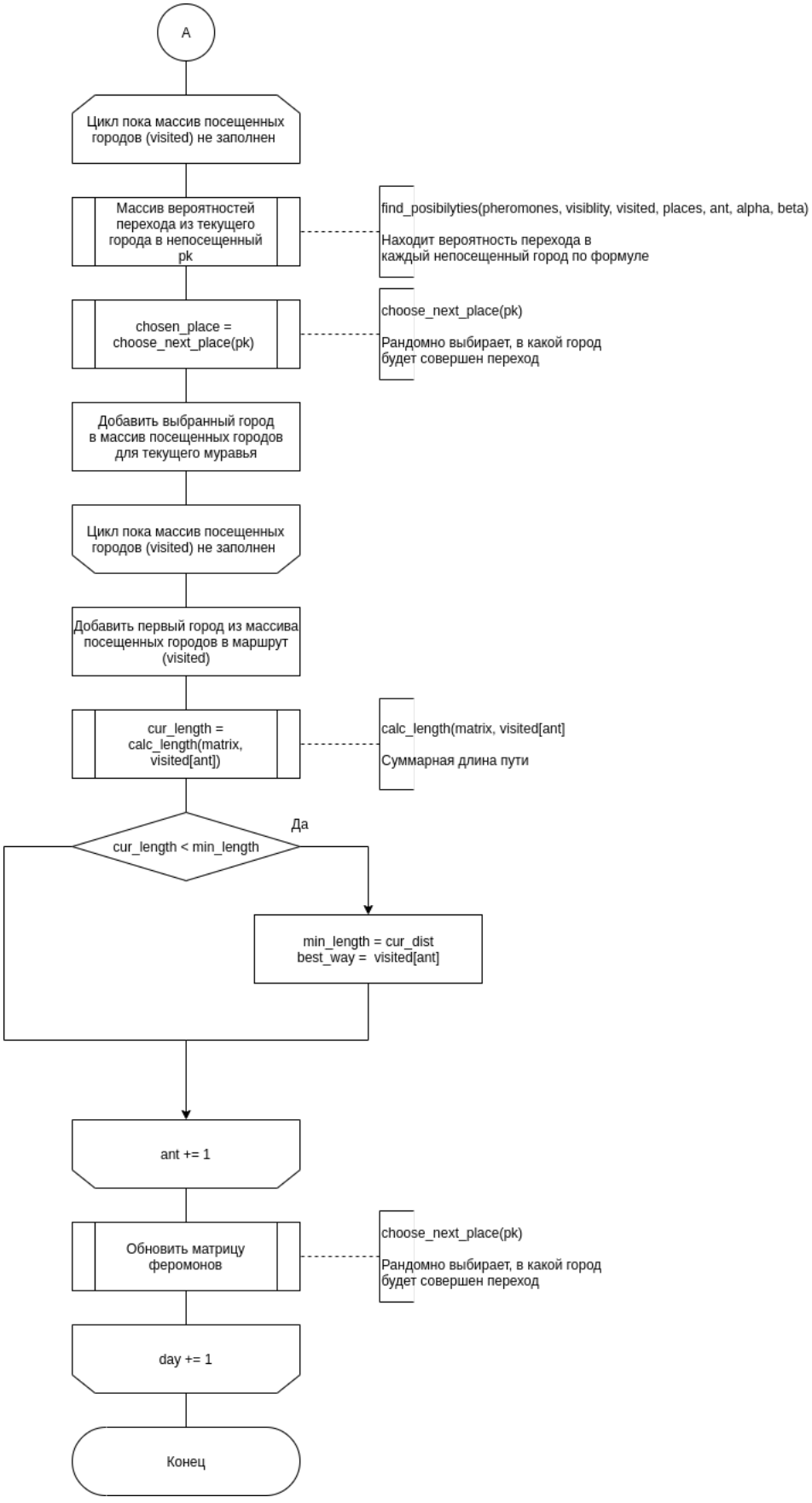

Рисунок 2.3 – Схема муравьиного алгоритма (часть 2)

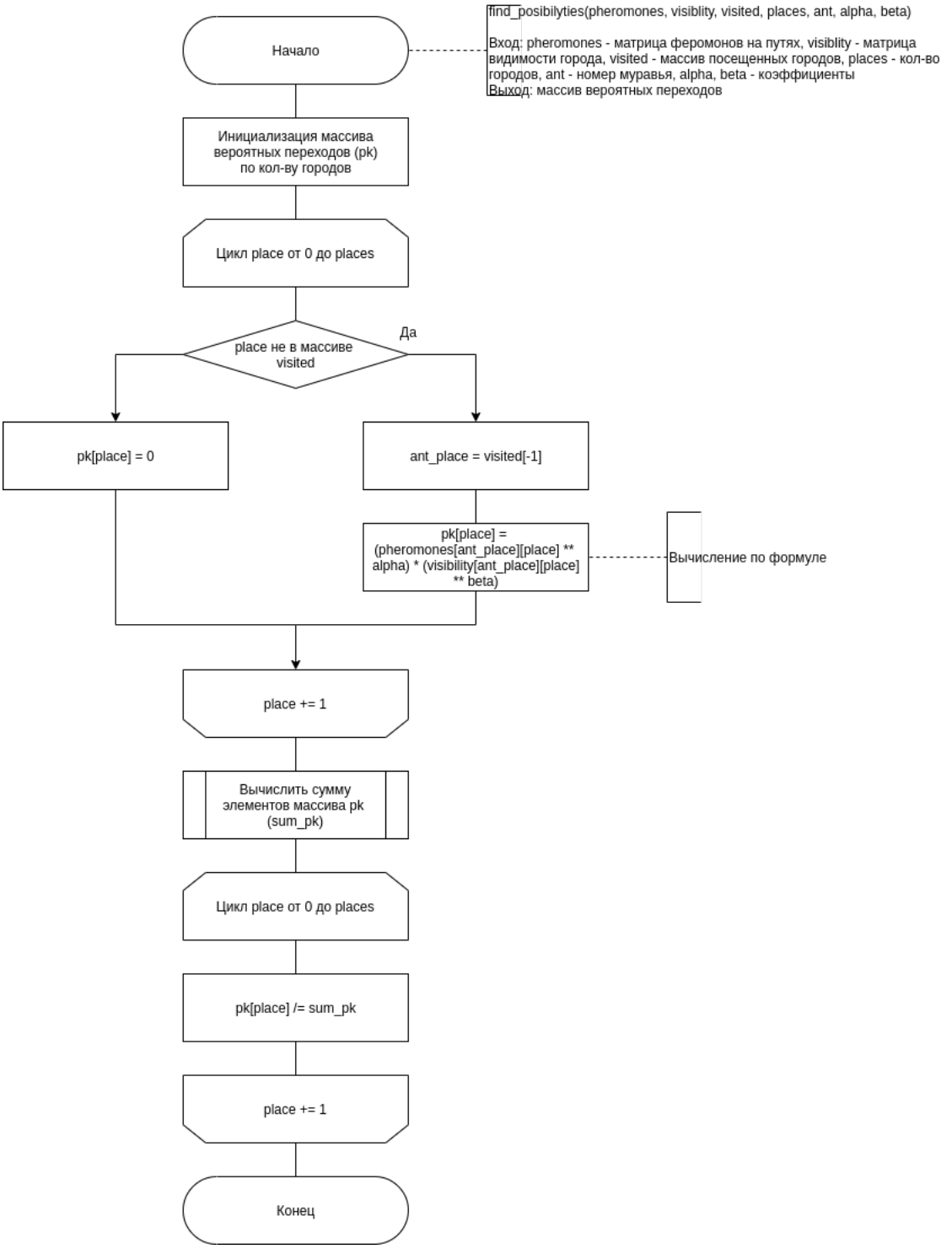

Рисунок 2.4 – Схема алгоритма нахождения массива вероятностных переходов в непосещенные города

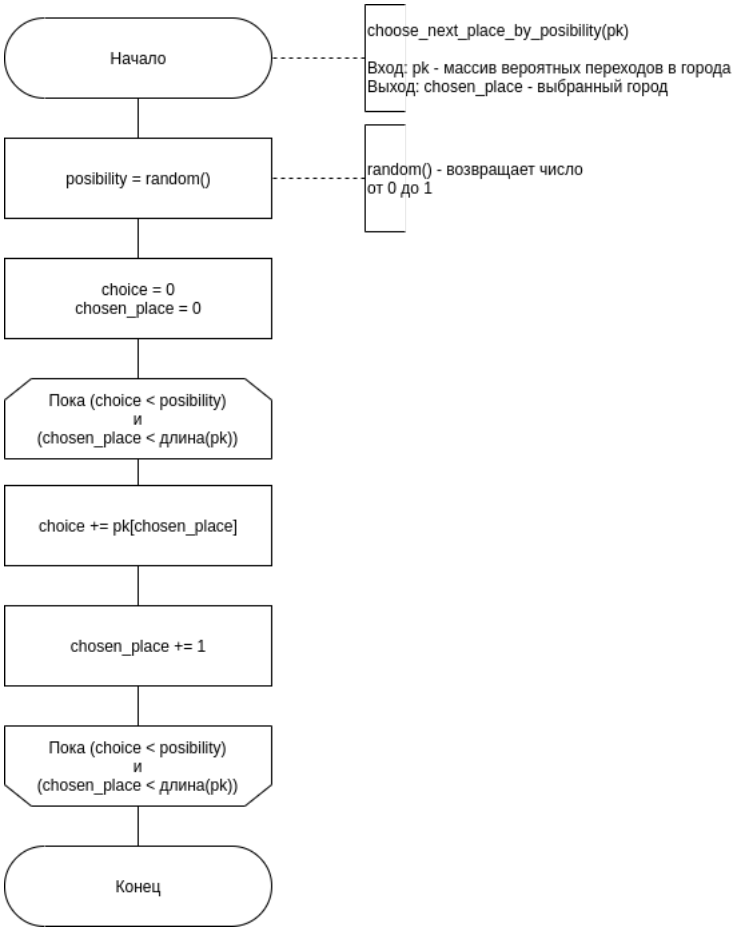

Рисунок 2.5 – Схема алгоритма нахождения следующего города на основании рандома

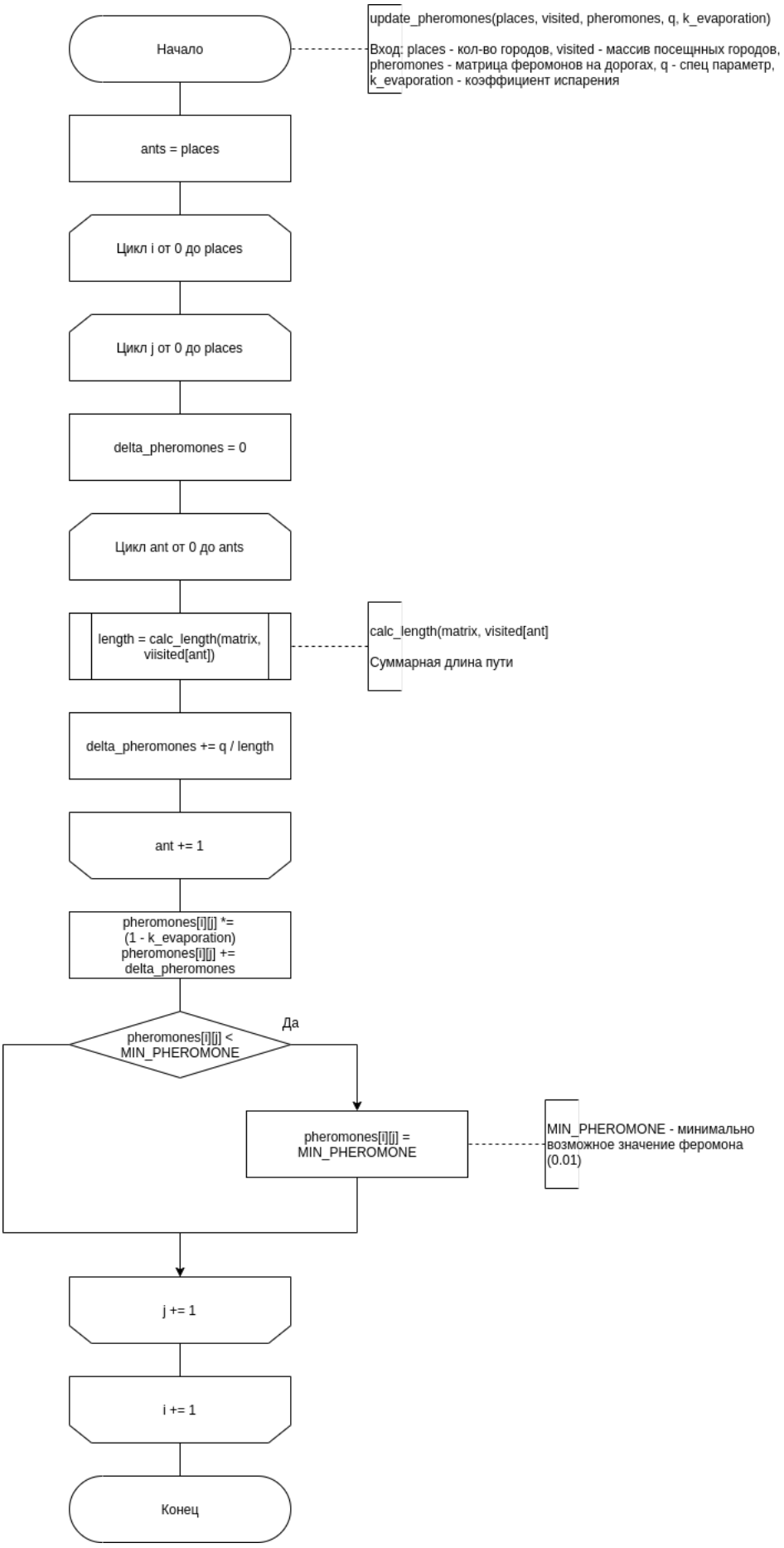

Рисунок 2.6 – Схема алгоритма обновления матрицы феромонов

2.4 Классы эквивалентности при тестировании

Для тестирования выделены классы эквивалентности, представленные ниже.

1. Неверно выбран пункт меню - не число или число, меньшее 0 или большее 8.
2. Неверно введены коэффициенты α, β, $evaporation\_koef$ - не число или число, меньшее 0.
3. Неверно введено кол-во дней - не число или число, меньшее 0.
4. Неверно введен размер матрицы - не число или число, меньшее 2.
5. Корректный ввод всех параметров.

2.5 Вывод

В данном разделе были построены схемы алгоритмов, рассматриваемых в лабораторной работе, были описаны классы эквивалентности для тестирования, структура программы.

3 Технологическая часть

В данном разделе будут рассмотрены средства реализации, а также представлены листинги алгоритма полного перебора путей и муравьиного алгоритма.

3.1 Средства реализации

В данной работе для реализации был выбран язык программирования *Python*[4]. В текущей лабораторной работе требуется замерить процессорное время для выполняемой программы, а также построить графики. Все эти инструменты присутствуют в выбранном языке программирования.

Время работы было замерено с помощью функции *process_time(...)* из библиотеки *time*[5].

3.2 Листинги кода

В листинге 3.1 представлен алгоритм полного перебора путей, а в листингах 3.2-3.5 - муравьиный алгоритм и дополнительные к нему функции.

Листинг 3.1 – Алгоритм полного перебора путей

```
def full_combinations(matrix, size):
    places = np.arange(size)
    places_combs = list()

    for combination in it.permutations(places):
        comb_arr = list(combination)

        places_combs.append(comb_arr)

    min_dist = float("inf")

    for i in range(len(places_combs)):
        places_combs[i].append(places_combs[i][0])

        cur_dist = 0

        for j in range(size):
            start_city = places_combs[i][j]
            end_city = places_combs[i][j + 1]

            cur_dist += matrix[start_city][end_city]

        if (cur_dist < min_dist):
            min_dist = cur_dist

            best_way = places_combs[i]

    return min_dist, best_way
```

Листинг 3.2 – Муравьиный алгоритм

```
def ant_algorythm(matrix, places, alpha, beta, k_evaporation, days):
    q = calc_q(matrix, places)

    best_way = []
    min_length = float("inf")

    pheromones = get_pheromones(places)
    visibility = get_visibility(matrix, places)

    ants = places

    for day in range(days):
        route = np.arange(places)

        visited = get_visited_places(route, ants)

        for ant in range(ants):
            while (len(visited[ant]) != ants):
                pk = find_posibilyties(pheromones, visibility,
                    visited, places, ant, alpha, beta)

                chosen_place = choose_next_place_by_posibility(pk)

                visited[ant].append(chosen_place - 1)

            visited[ant].append(visited[ant][0])

            cur_length = calc_length(matrix, visited[ant])

            if (cur_length < min_length):
                min_length = cur_length
                best_way = visited[ant]

        pheromones = update_pheromones(matrix, places, visited,
            pheromones, q, k_evaporation)

    return min_length, best_way
```

Листинг 3.3 – Алгоритм нахождения массива вероятностных переходов в непосещенные города

```
def find_posibilyties(pheromones, visibility, visited, places, ant,
    alpha, beta):
    pk = [0] * places

    for place in range(places):
        if place not in visited[ant]:
            ant_place = visited[ant][-1]

            pk[place] = pow(pheromones[ant_place][place], alpha) * \
                pow(visibility[ant_place][place], beta)

        else:
            pk[place] = 0

    sum_pk = sum(pk)

    for place in range(places):
        pk[place] /= sum_pk

    return pk
```

Листинг 3.4 – Алгоритм нахождения следующего города на основании рандома

```
def choose_next_place_by_posibility(pk):
    posibility = random()
    choice = 0
    chosen_place = 0

    while ((choice < posibility) and (chosen_place < len(pk))):
        choice += pk[chosen_place]
        chosen_place += 1

    return chosen_place
```

Листинг 3.5 – Алгоритм обновления матрицы феромонов

```
def update_pheromones(matrix, places, visited, pheromones, q,
    k_evaporation):
    ants = places

    for i in range(places):
        for j in range(places):
            delta_pheromones = 0

            for ant in range(ants):
                length = calc_length(matrix, visited[ant])
                delta_pheromones += q / length

            pheromones[i][j] *= (1 - k_evaporation)
            pheromones[i][j] += delta_pheromones

            if (pheromones[i][j] < MIN_PHEROMONE):
                pheromones[i][j] = MIN_PHEROMONE

    return pheromones
```

3.3 Сведения о модулях программы

Программа состоит из двух модулей:

- *main.py* - файл, содержащий меню программы, а также весь служебный код;
- *algorythms.py* - файл, содержащий реализацию алгоритма полного перебора и муравьиного алгоритма.

3.4 Функциональные тесты

В таблице 3.1 приведены тесты для функций программы. Тесты *для всех функций* 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]

3.5 Вывод

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

4 Исследовательская часть

В данном разделе будут приведены примеры работы программа, а также проведен сравнительный анализ алгоритмов при различных ситуациях на основе полученных данных.

4.1 Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование представлены далее:

- операционная система: Ubuntu 20.04.3 [6] Linux [7] x86_64;
- память: 8 GiB;
- процессор: Intel® Core™ i5-7300HQ CPU @ 2.50GHz [8].

При тестировании ноутбук был включен в сеть электропитания. Во время тестирования ноутбук был нагружен только встроенными приложениями окружения, а также системой тестирования.

4.2 Демонстрация работы программы

На рисунке 4.1 представлен результат работы программы для обоих алгоритмов – полного перебора и муравьиного.

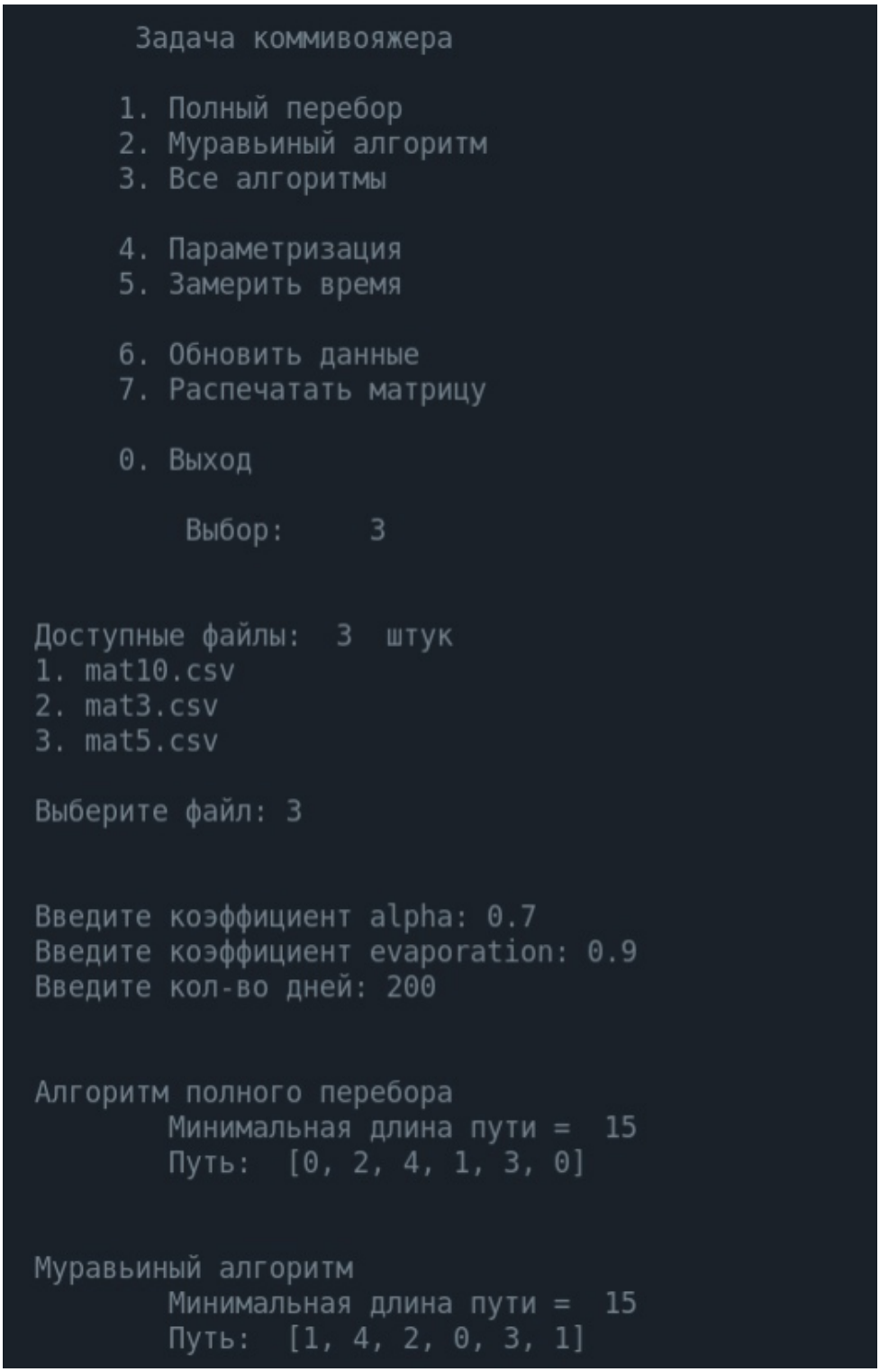

Рисунок 4.1 – Пример работы программы

4.3 Время выполнения алгоритмов

Как было сказано выше, используется функция замера процессорного времени *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

Размер	Полный перебор	Муравьиный
2	0.000130	0.019932
3	0.000138	0.031615
4	0.000104	0.044361
5	0.000420	0.089291
6	0.002390	0.152131
7	0.019703	0.254059
8	0.162850	0.398472
9	1.637611	0.594024
10	18.207853	0.857666

Также на рисунке 4.2 приведены графические результаты замеров.

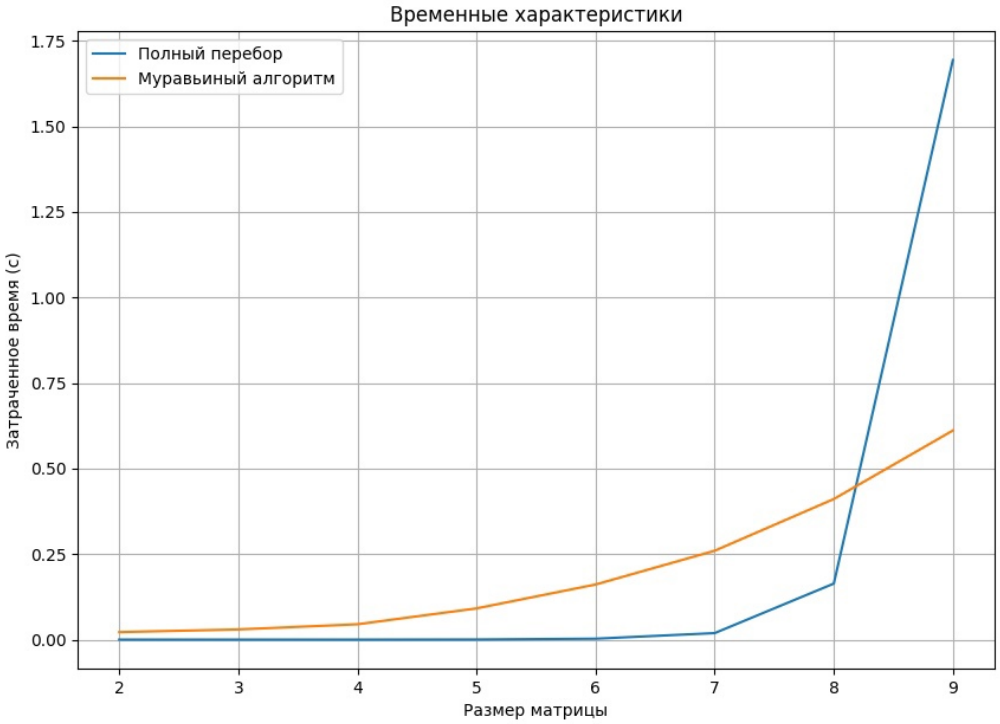

Рисунок 4.2 – Сравнение по времени алгоритмов полного перебора путей и муравьиного на разных размерах матриц

4.4 Постановка эксперимента

Автоматическая параметризация была проведена на двух классах данных – 4.1 и 4.2. Алгоритм будет запущен для всех значений $\alpha, \rho \in (0, 1)$.

Итоговая таблица значений параметризации будет состоять из следующих колонок:

- α - изменяющийся параметр;
- ρ - изменяющийся параметр;
- *days* - кол-во дней, изменяющийся параметр;
- *Result* - эталонный результат;
- *Mistake* - разность получившегося значения и эталонного значения на данных значениях параметров.

Цель эксперимента – определить комбинацию параметров, которые решают задачу наилучшим образом. Качество зависит от двух факторов:

- количества дней;
- погрешности измерений.

4.4.1 Класс данных 1

Класс данных 1 представляет собой матрицу смежности размером 9 элементов (небольшой разброс значений - от 1 до 2), которая представлена далее.

$$K_1 = \begin{pmatrix} 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 0 & 2 & 2 & 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 0 & 1 & 2 & 1 & 2 & 2 \\ 2 & 1 & 2 & 1 & 0 & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 2 & 2 & 0 & 1 & 1 & 2 \\ 1 & 2 & 1 & 1 & 2 & 1 & 0 & 2 & 2 \\ 1 & 1 & 2 & 2 & 1 & 1 & 2 & 0 & 2 \\ 2 & 1 & 2 & 2 & 1 & 2 & 2 & 2 & 0 \end{pmatrix} \tag{4.1}$$

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

Таблица 4.2 – Параметры для класса данных 1

α	ρ	Days	Result	Mistake
0.1	0.3	10	9	0
0.1	0.3	50	9	0
0.1	0.3	100	9	0
0.1	0.3	300	9	0
0.1	0.3	500	9	0
0.1	0.4	10	9	0
0.1	0.4	50	9	0
0.1	0.4	100	9	0
0.1	0.4	300	9	0
0.1	0.4	500	9	0
0.1	0.7	10	9	0
0.1	0.7	50	9	0
0.1	0.7	100	9	0
0.1	0.7	300	9	0
0.1	0.7	500	9	0
0.2	0.5	10	9	0
0.2	0.5	50	9	0
0.2	0.5	100	9	0

0.2	0.5	300	9	0
0.2	0.5	500	9	0
0.2	0.7	10	9	0
0.2	0.7	50	9	0
0.2	0.7	100	9	0
0.2	0.7	300	9	0
0.2	0.7	500	9	0
0.3	0.4	10	9	0
0.3	0.4	50	9	0
0.3	0.4	100	9	0
0.3	0.4	300	9	0
0.3	0.4	500	9	0
0.3	0.5	10	9	0
0.3	0.5	100	9	0
0.3	0.5	300	9	0
0.3	0.5	500	9	0
0.4	0.5	10	9	0
0.4	0.5	50	9	0
0.4	0.5	100	9	0
0.4	0.5	300	9	0
0.4	0.5	500	9	0
0.6	0.1	10	9	0
0.6	0.1	50	9	0
0.6	0.1	100	9	0
0.6	0.1	300	9	0
0.6	0.1	500	9	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.

$$K_1 = \begin{pmatrix} 0 & 9271 & 8511 & 2010 & 1983 & 7296 & 7289 & 3024 & 1011 \\ 9271 & 0 & 7731 & 4865 & 5494 & 6812 & 4755 & 7780 & 7641 \\ 8511 & 7731 & 0 & 1515 & 9297 & 7506 & 5781 & 5804 & 7334 \\ 2010 & 4865 & 1515 & 0 & 3662 & 9597 & 2876 & 8188 & 9227 \\ 1983 & 5494 & 9297 & 3662 & 0 & 8700 & 4754 & 7445 & 3834 \\ 7296 & 6812 & 7506 & 9597 & 8700 & 0 & 4216 & 5553 & 8215 \\ 7289 & 4755 & 5781 & 2876 & 4754 & 4216 & 0 & 4001 & 4715 \\ 3024 & 7780 & 5804 & 8188 & 7445 & 5553 & 4001 & 0 & 9522 \\ 1011 & 7641 & 7334 & 9227 & 3834 & 8215 & 4715 & 9522 & 0 \end{pmatrix} \quad (4.2)$$

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

Таблица 4.3 – Параметры для класса данных 2

α	ρ	Days	Result	Mistake
0.1	0.3	100	34192	0
0.1	0.3	300	34192	0
0.1	0.3	500	34192	0
0.1	0.7	100	34192	0
0.1	0.7	300	34192	0
0.1	0.7	500	34192	0
0.2	0.1	100	34192	0
0.2	0.1	300	34192	0
0.2	0.1	500	34192	0
0.2	0.2	100	34192	0
0.2	0.2	300	34192	0
0.2	0.2	500	34192	0
0.2	0.5	100	34192	0
0.2	0.5	300	34192	0
0.2	0.5	500	34192	0
0.2	0.8	100	34192	0
0.2	0.8	300	34192	0

0.2	0.8	500	34192	0
0.3	0.1	100	34192	0
0.3	0.1	300	34192	0
0.3	0.1	500	34192	0
0.3	0.2	5	34192	0
0.3	0.2	50	34192	0
0.3	0.2	100	34192	0
0.3	0.2	300	34192	0
0.3	0.2	500	34192	0
0.4	0.5	50	34192	0
0.4	0.5	300	34192	0
0.4	0.5	500	34192	0
0.5	0.2	100	34192	0
0.5	0.2	300	34192	0
0.5	0.2	500	34192	0
0.6	0.2	100	34192	0
0.6	0.2	300	34192	0
0.6	0.2	500	34192	0
0.6	0.3	300	34192	0
0.6	0.3	500	34192	0
0.6	0.4	100	34192	0
0.6	0.4	500	34192	0
0.6	0.5	100	34192	0
0.6	0.5	300	34192	0
0.6	0.5	500	34192	0

4.5 Вывод

В результате эксперимента было получно, что использование муравьиного алгоритма наиболее эффективно при больших размерах матриц. Так при размере матрицы, равном 2, муравьиный алгоритм медленее алгоритма полного перебора в 153 раза, а при размере матрицы, равном 9, муравьиный алгоритм

быстрее алгоритма полного перебора в раз, а при размере в 10 – уже в 21 раз. Следовательно, при рамзерах матриц больше 8 следует использовать муравьиный алгоритм, но стоит учитывать, что он может давать погрешности вычислений.

Также при проведении эксперимента с классами данных было получено, что на первом классе данных (4.1) муравьиный алгоритм лучше всего показывает себя при параметрах:

- $\alpha = 0.1, \rho = 0.3, 0.4, 0.7$;
- $\alpha = 0.2, \rho = 0.5, 0.7$;
- $\alpha = 0.3, \rho = 0.4, 0.5$;
- $\alpha = 0.4, \rho = 0.5$;
- $\alpha = 0.6, \rho = 0.1$.

Следовательно, для класса данных 1 (4.1) стоит использовать данные параметры.

Для класса данных 2 (4.2) было получено, что наилучшим образом алгоритм работает на значениях параметров, которые представлены далее:

- $\alpha = 0.1, \rho = 0.3, 0.7$;
- $\alpha = 0.2, \rho = 0.1, 0.2, 0.5, 0.8$;
- $\alpha = 0.3, \rho = 0.1, 0.2$;
- $\alpha = 0.4, \rho = 0.5$;
- $\alpha = 0.5, \rho = 0.2$;
- $\alpha = 0.6, \rho = 0.2, 0.3, 0.4$.

То есть, для второго класса данных (4.2) стоит использовать данные параметры.

Также во время исллледования было замечено – чем меньше α, тем меньше погрешностей возникает. При этом кол-во дней сильно влияет на отсутствие погрешностей: чем значение параметра $Days$ больше, тем меньше погрешностей.

Заключение

В результате исследования было определено, что муравьиный алгоритм стоит использовать при больших рамзерах матрицы (больше 8), при этом алгоритм полного перебора в 21 раз меньше при значении размера, равном 10, и чем больше размер – тем муравьиный алгоритм быстрее. Но при этом стоит учитывать, что муравьиный алгоритм имеет погрешности в вычислениях, которые можно устранить путем выбора правильного набора параметров. Так, было установлено то, что чем меньше значение параметра α, тем точнее работает муравьиный алгоритм. Также при точность увеличивается при увеличении параметра *Days*, но это влияет на время работы алгоритма.

Цель, которая была поставлена в начале лабораторной работы была достигнута, а также в ходе выполнения лабораторной работы были решены следующие задачи:

- были изучены алгоритмы решения задачи коммивояжера – полного перебора путей и муравьиный;
- были реализованы алгоритмы полного перебора, а также муравьиный алгоритм для поиска путей;
- проведен сравнительный анализ по времени алгоритмов на разном размере матриц;
- проведена параметризация муравьиного алгоритма, которая показала лучшие наборы параметров для работы алгоритма на определенных классах данных;
- подготовлен отчёт о лабораторной работе.

Список литературы

[1] Задача коммивояжёра [Электронный ресурс]. Режим доступа: http://mech.math.msu.su/~shvetz/54/inf/perl-problems/chCommisVoyageur.xhtml (дата обращения: 23.10.2021).

[2] Решаем задачу коммивояжёра простым перебором [Электронный ресурс]. Режим доступа: https://thecode.media/path-js/ (дата обращения: 23.11.2021).

[3] Муравьиные алгоритмы [Электронный ресурс]. Режим доступа: https://habr.com/ru/post/105302/ (дата обращения: 23.11.2021).

[4] Welcome to Python [Электронный ресурс]. Режим доступа: https://www.python.org (дата обращения: 23.11.2021).

[5] time — Time access and conversions [Электронный ресурс]. Режим доступа: https://docs.python.org/3/library/time.html#functions (дата обращения: 27.11.2021).

[6] Ubuntu 20.04.3 LTS (Focal Fossa) [Электронный ресурс]. Режим доступа: https://releases.ubuntu.com/20.04/ (дата обращения: 27.11.2021).

[7] Linux [Электронный ресурс]. Режим доступа: https://www.linux.org/forums/#linux-tutorials.122 (дата обращения: 27.11.2021).

[8] Процессор Intel® Core™ i5-7300HQ [Электронный ресурс]. Режим доступа: https://ark.intel.com/content/www/ru/ru/ark/products/97456/intel-core-i5-7300hq-processor-6m-cache-up-to-3-50-ghz.html (дата обращения: 25.11.2021).

Приложение 1

Таблица 4.4 – Параметризация для класса данных 1

α	ρ	Days	Result	Mistake
0.1	0.1	3	9	1
0.1	0.1	5	9	1
0.1	0.1	10	9	1
0.1	0.1	50	9	1
0.1	0.1	100	9	0
0.1	0.1	300	9	0
0.1	0.1	500	9	0
0.1	0.2	1	9	2
0.1	0.2	3	9	2
0.1	0.2	5	9	1
0.1	0.2	10	9	1
0.1	0.2	50	9	0
0.1	0.2	100	9	0
0.1	0.2	300	9	0
0.1	0.2	500	9	0
0.1	0.3	1	9	3
0.1	0.3	3	9	2
0.1	0.3	5	9	1
0.1	0.3	10	9	0
0.1	0.3	50	9	0
0.1	0.3	100	9	0
0.1	0.3	300	9	0
0.1	0.3	500	9	0
0.1	0.4	1	9	2
0.1	0.4	3	9	2
0.1	0.4	5	9	1
0.1	0.4	10	9	0
0.1	0.4	50	9	0

0.1	0.4	100	9	0
0.1	0.4	300	9	0
0.1	0.4	500	9	0
0.1	0.5	1	9	1
0.1	0.5	3	9	1
0.1	0.5	5	9	1
0.1	0.5	10	9	1
0.1	0.5	50	9	1
0.1	0.5	100	9	0
0.1	0.5	300	9	0
0.1	0.5	500	9	0
0.1	0.6	1	9	2
0.1	0.6	3	9	2
0.1	0.6	5	9	1
0.1	0.6	10	9	1
0.1	0.6	50	9	1
0.1	0.6	100	9	0
0.1	0.6	300	9	0
0.1	0.6	500	9	0
0.1	0.7	1	9	3
0.1	0.7	3	9	1
0.1	0.7	5	9	1
0.1	0.7	10	9	0
0.1	0.7	50	9	0
0.1	0.7	100	9	0
0.1	0.7	300	9	0
0.1	0.7	500	9	0
0.1	0.8	1	9	2
0.1	0.8	3	9	2
0.1	0.8	5	9	1
0.1	0.8	10	9	1
0.1	0.8	50	9	0
0.1	0.8	100	9	0

0.1	0.8	300	9	0
0.1	0.8	500	9	0
0.2	0.1	1	9	2
0.2	0.1	3	9	1
0.2	0.1	5	9	1
0.2	0.1	10	9	1
0.2	0.1	50	9	0
0.2	0.1	100	9	0
0.2	0.1	300	9	0
0.2	0.1	500	9	0
0.2	0.2	1	9	3
0.2	0.2	3	9	2
0.2	0.2	5	9	1
0.2	0.2	10	9	1
0.2	0.2	50	9	0
0.2	0.2	100	9	0
0.2	0.2	300	9	0
0.2	0.2	500	9	0
0.2	0.3	1	9	1
0.2	0.3	3	9	1
0.2	0.3	5	9	1
0.2	0.3	10	9	1
0.2	0.3	50	9	0
0.2	0.3	100	9	0
0.2	0.3	300	9	0
0.2	0.3	500	9	0
0.2	0.4	1	9	2
0.2	0.4	3	9	1
0.2	0.4	5	9	1
0.2	0.4	10	9	1
0.2	0.4	50	9	0
0.2	0.4	100	9	0
0.2	0.4	300	9	0

0.2	0.4	500	9	0
0.2	0.5	1	9	1
0.2	0.5	3	9	2
0.2	0.5	5	9	1
0.2	0.5	10	9	0
0.2	0.5	50	9	0
0.2	0.5	100	9	0
0.2	0.5	300	9	0
0.2	0.5	500	9	0
0.2	0.6	1	9	2
0.2	0.6	3	9	1
0.2	0.6	5	9	0
0.2	0.6	10	9	1
0.2	0.6	50	9	1
0.2	0.6	100	9	0
0.2	0.6	300	9	0
0.2	0.6	500	9	0
0.2	0.7	1	9	2
0.2	0.7	3	9	2
0.2	0.7	5	9	1
0.2	0.7	10	9	0
0.2	0.7	50	9	0
0.2	0.7	100	9	0
0.2	0.7	300	9	0
0.2	0.7	500	9	0
0.2	0.8	1	9	2
0.2	0.8	3	9	1
0.2	0.8	5	9	1
0.2	0.8	10	9	1
0.2	0.8	50	9	0
0.2	0.8	100	9	0
0.2	0.8	300	9	0
0.2	0.8	500	9	0

0.3	0.1	1	9	2
0.3	0.1	3	9	1
0.3	0.1	5	9	0
0.3	0.1	10	9	1
0.3	0.1	50	9	0
0.3	0.1	100	9	0
0.3	0.1	300	9	0
0.3	0.1	500	9	0
0.3	0.2	1	9	3
0.3	0.2	3	9	2
0.3	0.2	5	9	2
0.3	0.2	10	9	1
0.3	0.2	50	9	1
0.3	0.2	100	9	0
0.3	0.2	300	9	0
0.3	0.2	500	9	0
0.3	0.3	1	9	2
0.3	0.3	3	9	1
0.3	0.3	5	9	2
0.3	0.3	10	9	1
0.3	0.3	50	9	0
0.3	0.3	100	9	1
0.3	0.3	300	9	0
0.3	0.3	500	9	0
0.3	0.4	1	9	2
0.3	0.4	3	9	1
0.3	0.4	5	9	2
0.3	0.4	10	9	0
0.3	0.4	50	9	0
0.3	0.4	100	9	0
0.3	0.4	300	9	0
0.3	0.4	500	9	0
0.3	0.5	1	9	3

0.3	0.5	3	9	2
0.3	0.5	5	9	2
0.3	0.5	10	9	0
0.3	0.5	50	9	1
0.3	0.5	100	9	0
0.3	0.5	300	9	0
0.3	0.5	500	9	0
0.3	0.6	1	9	1
0.3	0.6	3	9	2
0.3	0.6	5	9	2
0.3	0.6	10	9	1
0.3	0.6	50	9	1
0.3	0.6	100	9	0
0.3	0.6	300	9	0
0.3	0.6	500	9	0
0.3	0.7	1	9	2
0.3	0.7	3	9	1
0.3	0.7	5	9	1
0.3	0.7	10	9	1
0.3	0.7	50	9	0
0.3	0.7	100	9	0
0.3	0.7	300	9	0
0.3	0.7	500	9	0
0.3	0.8	1	9	3
0.3	0.8	3	9	1
0.3	0.8	5	9	2
0.3	0.8	10	9	1
0.3	0.8	50	9	0
0.3	0.8	100	9	0
0.3	0.8	300	9	0
0.3	0.8	500	9	0
0.4	0.1	1	9	3
0.4	0.1	3	9	2

0.4	0.1	5	9	1
0.4	0.1	10	9	1
0.4	0.1	50	9	0
0.4	0.1	100	9	0
0.4	0.1	300	9	0
0.4	0.1	500	9	0
0.4	0.2	1	9	2
0.4	0.2	3	9	2
0.4	0.2	5	9	2
0.4	0.2	10	9	1
0.4	0.2	50	9	1
0.4	0.2	100	9	0
0.4	0.2	300	9	0
0.4	0.2	500	9	0
0.4	0.3	1	9	2
0.4	0.3	3	9	2
0.4	0.3	5	9	1
0.4	0.3	10	9	2
0.4	0.3	50	9	0
0.4	0.3	100	9	0
0.4	0.3	300	9	0
0.4	0.3	500	9	0
0.4	0.4	1	9	1
0.4	0.4	3	9	2
0.4	0.4	5	9	1
0.4	0.4	10	9	1
0.4	0.4	50	9	1
0.4	0.4	100	9	1
0.4	0.4	300	9	0
0.4	0.4	500	9	0
0.4	0.5	1	9	1
0.4	0.5	3	9	1
0.4	0.5	5	9	2

0.4	0.5	10	9	0
0.4	0.5	50	9	0
0.4	0.5	100	9	0
0.4	0.5	300	9	0
0.4	0.5	500	9	0
0.4	0.6	1	9	2
0.4	0.6	3	9	1
0.4	0.6	5	9	0
0.4	0.6	10	9	1
0.4	0.6	50	9	1
0.4	0.6	100	9	0
0.4	0.6	300	9	0
0.4	0.6	500	9	0
0.4	0.7	1	9	3
0.4	0.7	3	9	2
0.4	0.7	5	9	3
0.4	0.7	10	9	1
0.4	0.7	50	9	1
0.4	0.7	100	9	0
0.4	0.7	300	9	0
0.4	0.7	500	9	0
0.4	0.8	1	9	2
0.4	0.8	3	9	2
0.4	0.8	5	9	2
0.4	0.8	10	9	1
0.4	0.8	50	9	0
0.4	0.8	100	9	0
0.4	0.8	300	9	0
0.4	0.8	500	9	0
0.5	0.1	1	9	2
0.5	0.1	3	9	0
0.5	0.1	5	9	2
0.5	0.1	10	9	1

0.5	0.1	50	9	0
0.5	0.1	100	9	0
0.5	0.1	300	9	0
0.5	0.1	500	9	0
0.5	0.2	1	9	2
0.5	0.2	3	9	2
0.5	0.2	5	9	1
0.5	0.2	10	9	1
0.5	0.2	50	9	1
0.5	0.2	100	9	0
0.5	0.2	300	9	0
0.5	0.2	500	9	0
0.5	0.3	1	9	2
0.5	0.3	3	9	2
0.5	0.3	5	9	1
0.5	0.3	10	9	2
0.5	0.3	50	9	0
0.5	0.3	100	9	0
0.5	0.3	300	9	0
0.5	0.3	500	9	0
0.5	0.4	1	9	2
0.5	0.4	3	9	1
0.5	0.4	5	9	1
0.5	0.4	10	9	1
0.5	0.4	50	9	0
0.5	0.4	100	9	0
0.5	0.4	300	9	0
0.5	0.4	500	9	0
0.5	0.5	1	9	2
0.5	0.5	3	9	1
0.5	0.5	5	9	1
0.5	0.5	10	9	1
0.5	0.5	50	9	0

0.5	0.5	100	9	0
0.5	0.5	300	9	0
0.5	0.5	500	9	0
0.5	0.6	1	9	2
0.5	0.6	3	9	3
0.5	0.6	5	9	1
0.5	0.6	10	9	1
0.5	0.6	50	9	0
0.5	0.6	100	9	0
0.5	0.6	300	9	0
0.5	0.6	500	9	0
0.5	0.7	1	9	1
0.5	0.7	3	9	1
0.5	0.7	5	9	1
0.5	0.7	10	9	1
0.5	0.7	50	9	1
0.5	0.7	100	9	0
0.5	0.7	300	9	0
0.5	0.7	500	9	0
0.5	0.8	1	9	2
0.5	0.8	3	9	2
0.5	0.8	5	9	1
0.5	0.8	10	9	1
0.5	0.8	50	9	1
0.5	0.8	100	9	1
0.5	0.8	300	9	0
0.5	0.8	500	9	0
0.6	0.1	1	9	2
0.6	0.1	3	9	2
0.6	0.1	5	9	1
0.6	0.1	10	9	0
0.6	0.1	50	9	0
0.6	0.1	100	9	0

0.6	0.1	300	9	0
0.6	0.1	500	9	0
0.6	0.2	1	9	2
0.6	0.2	3	9	0
0.6	0.2	5	9	1
0.6	0.2	10	9	2
0.6	0.2	50	9	0
0.6	0.2	100	9	0
0.6	0.2	300	9	0
0.6	0.2	500	9	0
0.6	0.3	1	9	2
0.6	0.3	3	9	1
0.6	0.3	5	9	2
0.6	0.3	10	9	1
0.6	0.3	50	9	1
0.6	0.3	100	9	1
0.6	0.3	300	9	0
0.6	0.3	500	9	0
0.6	0.4	1	9	3
0.6	0.4	3	9	2
0.6	0.4	5	9	1
0.6	0.4	10	9	1
0.6	0.4	50	9	1
0.6	0.4	100	9	1
0.6	0.4	300	9	0
0.6	0.4	500	9	0
0.6	0.5	1	9	1
0.6	0.5	3	9	2
0.6	0.5	5	9	2
0.6	0.5	10	9	1
0.6	0.5	50	9	1
0.6	0.5	100	9	1
0.6	0.5	300	9	0

0.6	0.5	500	9	0
0.6	0.6	1	9	2
0.6	0.6	3	9	1
0.6	0.6	5	9	2
0.6	0.6	10	9	1
0.6	0.6	50	9	1
0.6	0.6	100	9	0
0.6	0.6	300	9	1
0.6	0.6	500	9	0
0.6	0.7	1	9	1
0.6	0.7	3	9	1
0.6	0.7	5	9	2
0.6	0.7	10	9	0
0.6	0.7	50	9	1
0.6	0.7	100	9	0
0.6	0.7	300	9	0
0.6	0.7	500	9	0
0.6	0.8	1	9	3
0.6	0.8	3	9	2
0.6	0.8	5	9	1
0.6	0.8	10	9	2
0.6	0.8	50	9	1
0.6	0.8	100	9	1
0.6	0.8	300	9	0
0.6	0.8	500	9	0
0.7	0.1	1	9	3
0.7	0.1	3	9	1
0.7	0.1	5	9	2
0.7	0.1	10	9	1
0.7	0.1	50	9	1
0.7	0.1	100	9	1
0.7	0.1	300	9	1
0.7	0.1	500	9	0

0.7	0.2	1	9	3
0.7	0.2	3	9	2
0.7	0.2	5	9	1
0.7	0.2	10	9	2
0.7	0.2	50	9	0
0.7	0.2	100	9	1
0.7	0.2	300	9	0
0.7	0.2	500	9	0
0.7	0.3	1	9	2
0.7	0.3	3	9	2
0.7	0.3	5	9	2
0.7	0.3	10	9	1
0.7	0.3	50	9	1
0.7	0.3	100	9	0
0.7	0.3	300	9	0
0.7	0.3	500	9	0
0.7	0.4	1	9	3
0.7	0.4	3	9	2
0.7	0.4	5	9	2
0.7	0.4	10	9	1
0.7	0.4	50	9	0
0.7	0.4	100	9	0
0.7	0.4	300	9	0
0.7	0.4	500	9	0
0.7	0.5	1	9	3
0.7	0.5	3	9	2
0.7	0.5	5	9	2
0.7	0.5	10	9	1
0.7	0.5	50	9	1
0.7	0.5	100	9	1
0.7	0.5	300	9	0
0.7	0.5	500	9	0
0.7	0.6	1	9	3

0.7	0.6	3	9	2
0.7	0.6	5	9	2
0.7	0.6	10	9	1
0.7	0.6	50	9	1
0.7	0.6	100	9	1
0.7	0.6	300	9	0
0.7	0.6	500	9	0
0.7	0.7	1	9	2
0.7	0.7	3	9	2
0.7	0.7	5	9	3
0.7	0.7	10	9	1
0.7	0.7	50	9	0
0.7	0.7	100	9	0
0.7	0.7	300	9	0
0.7	0.7	500	9	0
0.7	0.8	1	9	1
0.7	0.8	3	9	2
0.7	0.8	5	9	1
0.7	0.8	10	9	1
0.7	0.8	50	9	1
0.7	0.8	100	9	0
0.7	0.8	300	9	0
0.7	0.8	500	9	0
0.8	0.1	1	9	2
0.8	0.1	3	9	2
0.8	0.1	5	9	2
0.8	0.1	10	9	2
0.8	0.1	50	9	0
0.8	0.1	100	9	0
0.8	0.1	300	9	0
0.8	0.1	500	9	0
0.8	0.2	1	9	2
0.8	0.2	3	9	2

0.8	0.2	5	9	1
0.8	0.2	10	9	0
0.8	0.2	50	9	1
0.8	0.2	100	9	1
0.8	0.2	300	9	0
0.8	0.2	500	9	0
0.8	0.3	1	9	2
0.8	0.3	3	9	2
0.8	0.3	5	9	1
0.8	0.3	10	9	1
0.8	0.3	50	9	0
0.8	0.3	100	9	0
0.8	0.3	300	9	1
0.8	0.3	500	9	0
0.8	0.4	1	9	1
0.8	0.4	3	9	2
0.8	0.4	5	9	1
0.8	0.4	10	9	1
0.8	0.4	50	9	1
0.8	0.4	100	9	0
0.8	0.4	300	9	1
0.8	0.4	500	9	0
0.8	0.5	1	9	4
0.8	0.5	3	9	2
0.8	0.5	5	9	1
0.8	0.5	10	9	2
0.8	0.5	50	9	1
0.8	0.5	100	9	0
0.8	0.5	300	9	0
0.8	0.5	500	9	0
0.8	0.6	1	9	3
0.8	0.6	3	9	2
0.8	0.6	5	9	0

0.8	0.6	10	9	1
0.8	0.6	50	9	0
0.8	0.6	100	9	1
0.8	0.6	300	9	0
0.8	0.6	500	9	0
0.8	0.7	1	9	3
0.8	0.7	3	9	2
0.8	0.7	5	9	1
0.8	0.7	10	9	2
0.8	0.7	50	9	1
0.8	0.7	100	9	0
0.8	0.7	300	9	0
0.8	0.7	500	9	0
0.8	0.8	1	9	2
0.8	0.8	3	9	1
0.8	0.8	5	9	1
0.8	0.8	10	9	2
0.8	0.8	50	9	1
0.8	0.8	100	9	1
0.8	0.8	300	9	0
0.8	0.8	500	9	0
0.9	0.1	1	9	2
0.9	0.1	3	9	2
0.9	0.1	5	9	1
0.9	0.1	10	9	1
0.9	0.1	50	9	1
0.9	0.1	100	9	1
0.9	0.1	300	9	0
0.9	0.1	500	9	0
0.9	0.2	1	9	3
0.9	0.2	3	9	2
0.9	0.2	5	9	1
0.9	0.2	10	9	2

0.9	0.2	50	9	1
0.9	0.2	100	9	1
0.9	0.2	300	9	0
0.9	0.2	500	9	1
0.9	0.3	1	9	2
0.9	0.3	3	9	2
0.9	0.3	5	9	2
0.9	0.3	10	9	2
0.9	0.3	50	9	1
0.9	0.3	100	9	0
0.9	0.3	300	9	1
0.9	0.3	500	9	0
0.9	0.4	1	9	2
0.9	0.4	3	9	2
0.9	0.4	5	9	2
0.9	0.4	10	9	1
0.9	0.4	50	9	0
0.9	0.4	100	9	1
0.9	0.4	300	9	1
0.9	0.4	500	9	0
0.9	0.5	1	9	3
0.9	0.5	3	9	2
0.9	0.5	5	9	2
0.9	0.5	10	9	2
0.9	0.5	50	9	1
0.9	0.5	100	9	1
0.9	0.5	300	9	0
0.9	0.5	500	9	0
0.9	0.6	1	9	3
0.9	0.6	3	9	2
0.9	0.6	5	9	2
0.9	0.6	10	9	1
0.9	0.6	50	9	1

0.9	0.6	100	9	1
0.9	0.6	300	9	0
0.9	0.6	500	9	0
0.9	0.7	1	9	1
0.9	0.7	3	9	2
0.9	0.7	5	9	1
0.9	0.7	10	9	0
0.9	0.7	50	9	1
0.9	0.7	100	9	1
0.9	0.7	300	9	0
0.9	0.7	500	9	0
0.9	0.8	1	9	3
0.9	0.8	3	9	1
0.9	0.8	5	9	1
0.9	0.8	10	9	2
0.9	0.8	50	9	1
0.9	0.8	100	9	1
0.9	0.8	300	9	0
0.9	0.8	500	9	0

Приложение 2

Таблица 4.5 – Параметризация для класса данных 2

α	ρ	Days	Result	Mistake
0.1	0.1	3	34192	3476
0.1	0.1	5	34192	4739
0.1	0.1	10	34192	1376
0.1	0.1	50	34192	505
0.1	0.1	100	34192	505
0.1	0.1	300	34192	0
0.1	0.1	500	34192	394
0.1	0.2	1	34192	5312
0.1	0.2	3	34192	0
0.1	0.2	5	34192	1062
0.1	0.2	10	34192	1487
0.1	0.2	50	34192	1487
0.1	0.2	100	34192	394
0.1	0.2	300	34192	0
0.1	0.2	500	34192	0
0.1	0.3	1	34192	5994
0.1	0.3	3	34192	5092
0.1	0.3	5	34192	964
0.1	0.3	10	34192	3043
0.1	0.3	50	34192	394
0.1	0.3	100	34192	0
0.1	0.3	300	34192	0
0.1	0.3	500	34192	0
0.1	0.4	1	34192	5617
0.1	0.4	3	34192	3636
0.1	0.4	5	34192	2789
0.1	0.4	10	34192	2918
0.1	0.4	50	34192	505

0.1	0.4	100	34192	1101
0.1	0.4	300	34192	0
0.1	0.4	500	34192	0
0.1	0.5	1	34192	8392
0.1	0.5	3	34192	2061
0.1	0.5	5	34192	1614
0.1	0.5	10	34192	1819
0.1	0.5	50	34192	394
0.1	0.5	100	34192	0
0.1	0.5	300	34192	394
0.1	0.5	500	34192	0
0.1	0.6	1	34192	1881
0.1	0.6	3	34192	4288
0.1	0.6	5	34192	1819
0.1	0.6	10	34192	4626
0.1	0.6	50	34192	394
0.1	0.6	100	34192	394
0.1	0.6	300	34192	0
0.1	0.6	500	34192	0
0.1	0.7	1	34192	10326
0.1	0.7	3	34192	6023
0.1	0.7	5	34192	2918
0.1	0.7	10	34192	964
0.1	0.7	50	34192	1948
0.1	0.7	100	34192	0
0.1	0.7	300	34192	0
0.1	0.7	500	34192	0
0.1	0.8	1	34192	964
0.1	0.8	3	34192	5156
0.1	0.8	5	34192	2625
0.1	0.8	10	34192	1614
0.1	0.8	50	34192	1274
0.1	0.8	100	34192	394

0.1	0.8	300	34192	0
0.1	0.8	500	34192	0
0.2	0.1	1	34192	10079
0.2	0.1	3	34192	3476
0.2	0.1	5	34192	1618
0.2	0.1	10	34192	1618
0.2	0.1	50	34192	1062
0.2	0.1	100	34192	0
0.2	0.1	300	34192	0
0.2	0.1	500	34192	0
0.2	0.2	1	34192	1928
0.2	0.2	3	34192	6474
0.2	0.2	5	34192	2372
0.2	0.2	10	34192	1614
0.2	0.2	50	34192	394
0.2	0.2	100	34192	0
0.2	0.2	300	34192	0
0.2	0.2	500	34192	0
0.2	0.3	1	34192	8194
0.2	0.3	3	34192	6747
0.2	0.3	5	34192	3513
0.2	0.3	10	34192	394
0.2	0.3	50	34192	1274
0.2	0.3	100	34192	0
0.2	0.3	300	34192	505
0.2	0.3	500	34192	394
0.2	0.4	1	34192	5041
0.2	0.4	3	34192	3946
0.2	0.4	5	34192	1109
0.2	0.4	10	34192	2957
0.2	0.4	50	34192	1062
0.2	0.4	100	34192	1376
0.2	0.4	300	34192	0

0.2	0.4	500	34192	0
0.2	0.5	1	34192	2791
0.2	0.5	3	34192	3890
0.2	0.5	5	34192	394
0.2	0.5	10	34192	2789
0.2	0.5	50	34192	394
0.2	0.5	100	34192	0
0.2	0.5	300	34192	0
0.2	0.5	500	34192	0
0.2	0.6	1	34192	6474
0.2	0.6	3	34192	8835
0.2	0.6	5	34192	2451
0.2	0.6	10	34192	2957
0.2	0.6	50	34192	1487
0.2	0.6	100	34192	964
0.2	0.6	300	34192	0
0.2	0.6	500	34192	0
0.2	0.7	1	34192	7490
0.2	0.7	3	34192	4383
0.2	0.7	5	34192	2957
0.2	0.7	10	34192	394
0.2	0.7	50	34192	1881
0.2	0.7	100	34192	394
0.2	0.7	300	34192	0
0.2	0.7	500	34192	0
0.2	0.8	1	34192	5399
0.2	0.8	3	34192	2887
0.2	0.8	5	34192	1062
0.2	0.8	10	34192	3150
0.2	0.8	50	34192	1571
0.2	0.8	100	34192	0
0.2	0.8	300	34192	0
0.2	0.8	500	34192	0

0.3	0.1	1	34192	8753
0.3	0.1	3	34192	1376
0.3	0.1	5	34192	5939
0.3	0.1	10	34192	1614
0.3	0.1	50	34192	964
0.3	0.1	100	34192	0
0.3	0.1	300	34192	0
0.3	0.1	500	34192	0
0.3	0.2	1	34192	1618
0.3	0.2	3	34192	3443
0.3	0.2	5	34192	0
0.3	0.2	10	34192	1643
0.3	0.2	50	34192	0
0.3	0.2	100	34192	0
0.3	0.2	300	34192	0
0.3	0.2	500	34192	0
0.3	0.3	1	34192	7319
0.3	0.3	3	34192	3705
0.3	0.3	5	34192	1109
0.3	0.3	10	34192	1109
0.3	0.3	50	34192	964
0.3	0.3	100	34192	964
0.3	0.3	300	34192	505
0.3	0.3	500	34192	0
0.3	0.4	1	34192	7998
0.3	0.4	3	34192	1819
0.3	0.4	5	34192	1487
0.3	0.4	10	34192	1948
0.3	0.4	50	34192	1881
0.3	0.4	100	34192	1062
0.3	0.4	300	34192	505
0.3	0.4	500	34192	0
0.3	0.5	1	34192	3043

0.3	0.5	3	34192	4226
0.3	0.5	5	34192	1948
0.3	0.5	10	34192	2372
0.3	0.5	50	34192	1571
0.3	0.5	100	34192	0
0.3	0.5	300	34192	0
0.3	0.5	500	34192	0
0.3	0.6	1	34192	9152
0.3	0.6	3	34192	7008
0.3	0.6	5	34192	1376
0.3	0.6	10	34192	4524
0.3	0.6	50	34192	1819
0.3	0.6	100	34192	1062
0.3	0.6	300	34192	0
0.3	0.6	500	34192	0
0.3	0.7	1	34192	8773
0.3	0.7	3	34192	8926
0.3	0.7	5	34192	1109
0.3	0.7	10	34192	4228
0.3	0.7	50	34192	1618
0.3	0.7	100	34192	0
0.3	0.7	300	34192	394
0.3	0.7	500	34192	0
0.3	0.8	1	34192	6534
0.3	0.8	3	34192	6118
0.3	0.8	5	34192	3705
0.3	0.8	10	34192	4048
0.3	0.8	50	34192	1856
0.3	0.8	100	34192	0
0.3	0.8	300	34192	505
0.3	0.8	500	34192	0
0.4	0.1	1	34192	4263
0.4	0.1	3	34192	1614

0.4	0.1	5	34192	3544
0.4	0.1	10	34192	1109
0.4	0.1	50	34192	0
0.4	0.1	100	34192	0
0.4	0.1	300	34192	505
0.4	0.1	500	34192	394
0.4	0.2	1	34192	9255
0.4	0.2	3	34192	7349
0.4	0.2	5	34192	6474
0.4	0.2	10	34192	3558
0.4	0.2	50	34192	1274
0.4	0.2	100	34192	964
0.4	0.2	300	34192	0
0.4	0.2	500	34192	0
0.4	0.3	1	34192	8006
0.4	0.3	3	34192	4697
0.4	0.3	5	34192	3878
0.4	0.3	10	34192	3443
0.4	0.3	50	34192	1062
0.4	0.3	100	34192	1819
0.4	0.3	300	34192	0
0.4	0.3	500	34192	394
0.4	0.4	1	34192	11660
0.4	0.4	3	34192	6474
0.4	0.4	5	34192	3306
0.4	0.4	10	34192	4702
0.4	0.4	50	34192	1571
0.4	0.4	100	34192	505
0.4	0.4	300	34192	394
0.4	0.4	500	34192	394
0.4	0.5	1	34192	4793
0.4	0.5	3	34192	4247
0.4	0.5	5	34192	2957

0.4	0.5	10	34192	1062
0.4	0.5	50	34192	0
0.4	0.5	100	34192	505
0.4	0.5	300	34192	0
0.4	0.5	500	34192	0
0.4	0.6	1	34192	10241
0.4	0.6	3	34192	1571
0.4	0.6	5	34192	3014
0.4	0.6	10	34192	1819
0.4	0.6	50	34192	1928
0.4	0.6	100	34192	505
0.4	0.6	300	34192	505
0.4	0.6	500	34192	0
0.4	0.7	1	34192	14336
0.4	0.7	3	34192	2789
0.4	0.7	5	34192	1643
0.4	0.7	10	34192	2451
0.4	0.7	50	34192	505
0.4	0.7	100	34192	964
0.4	0.7	300	34192	0
0.4	0.7	500	34192	394
0.4	0.8	1	34192	2912
0.4	0.8	3	34192	3392
0.4	0.8	5	34192	3624
0.4	0.8	10	34192	1928
0.4	0.8	50	34192	1487
0.4	0.8	100	34192	394
0.4	0.8	300	34192	0
0.4	0.8	500	34192	0
0.5	0.1	1	34192	6093
0.5	0.1	3	34192	9491
0.5	0.1	5	34192	3111
0.5	0.1	10	34192	4546

0.5	0.1	50	34192	1376
0.5	0.1	100	34192	1571
0.5	0.1	300	34192	0
0.5	0.1	500	34192	0
0.5	0.2	1	34192	2451
0.5	0.2	3	34192	5351
0.5	0.2	5	34192	1643
0.5	0.2	10	34192	1376
0.5	0.2	50	34192	1101
0.5	0.2	100	34192	0
0.5	0.2	300	34192	0
0.5	0.2	500	34192	0
0.5	0.3	1	34192	3504
0.5	0.3	3	34192	4938
0.5	0.3	5	34192	3111
0.5	0.3	10	34192	2912
0.5	0.3	50	34192	1755
0.5	0.3	100	34192	1376
0.5	0.3	300	34192	0
0.5	0.3	500	34192	0
0.5	0.4	1	34192	10923
0.5	0.4	3	34192	3757
0.5	0.4	5	34192	4337
0.5	0.4	10	34192	4706
0.5	0.4	50	34192	1881
0.5	0.4	100	34192	1062
0.5	0.4	300	34192	394
0.5	0.4	500	34192	0
0.5	0.5	1	34192	12953
0.5	0.5	3	34192	5647
0.5	0.5	5	34192	4174
0.5	0.5	10	34192	2791
0.5	0.5	50	34192	0

0.5	0.5	100	34192	1109
0.5	0.5	300	34192	394
0.5	0.5	500	34192	0
0.5	0.6	1	34192	6769
0.5	0.6	3	34192	1274
0.5	0.6	5	34192	2918
0.5	0.6	10	34192	4263
0.5	0.6	50	34192	394
0.5	0.6	100	34192	394
0.5	0.6	300	34192	1062
0.5	0.6	500	34192	0
0.5	0.7	1	34192	7796
0.5	0.7	3	34192	7016
0.5	0.7	5	34192	7199
0.5	0.7	10	34192	2387
0.5	0.7	50	34192	1062
0.5	0.7	100	34192	1062
0.5	0.7	300	34192	0
0.5	0.7	500	34192	0
0.5	0.8	1	34192	6495
0.5	0.8	3	34192	4383
0.5	0.8	5	34192	0
0.5	0.8	10	34192	5225
0.5	0.8	50	34192	2451
0.5	0.8	100	34192	1856
0.5	0.8	300	34192	0
0.5	0.8	500	34192	0
0.6	0.1	1	34192	9525
0.6	0.1	3	34192	4918
0.6	0.1	5	34192	5598
0.6	0.1	10	34192	2918
0.6	0.1	50	34192	3624
0.6	0.1	100	34192	2625

0.6	0.1	300	34192	505
0.6	0.1	500	34192	394
0.6	0.2	1	34192	5389
0.6	0.2	3	34192	2372
0.6	0.2	5	34192	3043
0.6	0.2	10	34192	3543
0.6	0.2	50	34192	1614
0.6	0.2	100	34192	0
0.6	0.2	300	34192	0
0.6	0.2	500	34192	0
0.6	0.3	1	34192	4706
0.6	0.3	3	34192	5246
0.6	0.3	5	34192	1928
0.6	0.3	10	34192	1881
0.6	0.3	50	34192	1571
0.6	0.3	100	34192	1101
0.6	0.3	300	34192	0
0.6	0.3	500	34192	0
0.6	0.4	1	34192	8825
0.6	0.4	3	34192	7121
0.6	0.4	5	34192	1881
0.6	0.4	10	34192	1618
0.6	0.4	50	34192	394
0.6	0.4	100	34192	0
0.6	0.4	300	34192	394
0.6	0.4	500	34192	0
0.6	0.5	1	34192	8190
0.6	0.5	3	34192	5838
0.6	0.5	5	34192	5161
0.6	0.5	10	34192	6575
0.6	0.5	50	34192	1376
0.6	0.5	100	34192	0
0.6	0.5	300	34192	0

0.6	0.5	500	34192	0
0.6	0.6	1	34192	3029
0.6	0.6	3	34192	5389
0.6	0.6	5	34192	4778
0.6	0.6	10	34192	3566
0.6	0.6	50	34192	964
0.6	0.6	100	34192	1614
0.6	0.6	300	34192	0
0.6	0.6	500	34192	505
0.6	0.7	1	34192	9846
0.6	0.7	3	34192	8610
0.6	0.7	5	34192	5544
0.6	0.7	10	34192	4247
0.6	0.7	50	34192	505
0.6	0.7	100	34192	0
0.6	0.7	300	34192	505
0.6	0.7	500	34192	0
0.6	0.8	1	34192	14502
0.6	0.8	3	34192	6162
0.6	0.8	5	34192	4383
0.6	0.8	10	34192	3002
0.6	0.8	50	34192	2061
0.6	0.8	100	34192	1571
0.6	0.8	300	34192	505
0.6	0.8	500	34192	505
0.7	0.1	1	34192	12035
0.7	0.1	3	34192	8649
0.7	0.1	5	34192	3844
0.7	0.1	10	34192	3014
0.7	0.1	50	34192	1376
0.7	0.1	100	34192	1928
0.7	0.1	300	34192	1274
0.7	0.1	500	34192	505

0.7	0.2	1	34192	5195
0.7	0.2	3	34192	4702
0.7	0.2	5	34192	8292
0.7	0.2	10	34192	4485
0.7	0.2	50	34192	1643
0.7	0.2	100	34192	0
0.7	0.2	300	34192	1614
0.7	0.2	500	34192	0
0.7	0.3	1	34192	6823
0.7	0.3	3	34192	2789
0.7	0.3	5	34192	1948
0.7	0.3	10	34192	3878
0.7	0.3	50	34192	505
0.7	0.3	100	34192	0
0.7	0.3	300	34192	1109
0.7	0.3	500	34192	505
0.7	0.4	1	34192	7758
0.7	0.4	3	34192	3532
0.7	0.4	5	34192	1881
0.7	0.4	10	34192	2918
0.7	0.4	50	34192	394
0.7	0.4	100	34192	1376
0.7	0.4	300	34192	394
0.7	0.4	500	34192	964
0.7	0.5	1	34192	10241
0.7	0.5	3	34192	7957
0.7	0.5	5	34192	3705
0.7	0.5	10	34192	4234
0.7	0.5	50	34192	2789
0.7	0.5	100	34192	1376
0.7	0.5	300	34192	1274
0.7	0.5	500	34192	0
0.7	0.6	1	34192	10820

0.7	0.6	3	34192	2372
0.7	0.6	5	34192	3767
0.7	0.6	10	34192	3392
0.7	0.6	50	34192	2789
0.7	0.6	100	34192	1062
0.7	0.6	300	34192	394
0.7	0.6	500	34192	1062
0.7	0.7	1	34192	11745
0.7	0.7	3	34192	8092
0.7	0.7	5	34192	4585
0.7	0.7	10	34192	4477
0.7	0.7	50	34192	1274
0.7	0.7	100	34192	1487
0.7	0.7	300	34192	0
0.7	0.7	500	34192	0
0.7	0.8	1	34192	12321
0.7	0.8	3	34192	8090
0.7	0.8	5	34192	5313
0.7	0.8	10	34192	4920
0.7	0.8	50	34192	2877
0.7	0.8	100	34192	2451
0.7	0.8	300	34192	505
0.7	0.8	500	34192	0
0.8	0.1	1	34192	15170
0.8	0.1	3	34192	6650
0.8	0.1	5	34192	8632
0.8	0.1	10	34192	3494
0.8	0.1	50	34192	2387
0.8	0.1	100	34192	1274
0.8	0.1	300	34192	964
0.8	0.1	500	34192	394
0.8	0.2	1	34192	10174
0.8	0.2	3	34192	2957

0.8	0.2	5	34192	4234
0.8	0.2	10	34192	3851
0.8	0.2	50	34192	1062
0.8	0.2	100	34192	1101
0.8	0.2	300	34192	394
0.8	0.2	500	34192	505
0.8	0.3	1	34192	11865
0.8	0.3	3	34192	2453
0.8	0.3	5	34192	4048
0.8	0.3	10	34192	4139
0.8	0.3	50	34192	505
0.8	0.3	100	34192	3014
0.8	0.3	300	34192	1062
0.8	0.3	500	34192	0
0.8	0.4	1	34192	5920
0.8	0.4	3	34192	5734
0.8	0.4	5	34192	7252
0.8	0.4	10	34192	4485
0.8	0.4	50	34192	1571
0.8	0.4	100	34192	1109
0.8	0.4	300	34192	394
0.8	0.4	500	34192	505
0.8	0.5	1	34192	4697
0.8	0.5	3	34192	3119
0.8	0.5	5	34192	3574
0.8	0.5	10	34192	6995
0.8	0.5	50	34192	3532
0.8	0.5	100	34192	1487
0.8	0.5	300	34192	1062
0.8	0.5	500	34192	0
0.8	0.6	1	34192	14377
0.8	0.6	3	34192	7758
0.8	0.6	5	34192	3844

0.8	0.6	10	34192	5734
0.8	0.6	50	34192	1571
0.8	0.6	100	34192	1376
0.8	0.6	300	34192	505
0.8	0.6	500	34192	964
0.8	0.7	1	34192	11213
0.8	0.7	3	34192	4915
0.8	0.7	5	34192	5348
0.8	0.7	10	34192	5496
0.8	0.7	50	34192	2789
0.8	0.7	100	34192	394
0.8	0.7	300	34192	1376
0.8	0.7	500	34192	0
0.8	0.8	1	34192	8398
0.8	0.8	3	34192	2887
0.8	0.8	5	34192	3878
0.8	0.8	10	34192	4637
0.8	0.8	50	34192	2625
0.8	0.8	100	34192	1881
0.8	0.8	300	34192	964
0.8	0.8	500	34192	0
0.9	0.1	1	34192	9919
0.9	0.1	3	34192	5348
0.9	0.1	5	34192	5092
0.9	0.1	10	34192	7604
0.9	0.1	50	34192	2387
0.9	0.1	100	34192	2912
0.9	0.1	300	34192	1571
0.9	0.1	500	34192	0
0.9	0.2	1	34192	11649
0.9	0.2	3	34192	5058
0.9	0.2	5	34192	5246
0.9	0.2	10	34192	2957

0.9	0.2	50	34192	1487
0.9	0.2	100	34192	394
0.9	0.2	300	34192	0
0.9	0.2	500	34192	505
0.9	0.3	1	34192	14873
0.9	0.3	3	34192	5166
0.9	0.3	5	34192	4990
0.9	0.3	10	34192	3543
0.9	0.3	50	34192	1109
0.9	0.3	100	34192	1376
0.9	0.3	300	34192	0
0.9	0.3	500	34192	505
0.9	0.4	1	34192	12211
0.9	0.4	3	34192	7207
0.9	0.4	5	34192	11281
0.9	0.4	10	34192	8874
0.9	0.4	50	34192	1618
0.9	0.4	100	34192	964
0.9	0.4	300	34192	1101
0.9	0.4	500	34192	1376
0.9	0.5	1	34192	8673
0.9	0.5	3	34192	2887
0.9	0.5	5	34192	8356
0.9	0.5	10	34192	5000
0.9	0.5	50	34192	3150
0.9	0.5	100	34192	1614
0.9	0.5	300	34192	1614
0.9	0.5	500	34192	505
0.9	0.6	1	34192	12873
0.9	0.6	3	34192	3435
0.9	0.6	5	34192	3757
0.9	0.6	10	34192	6633
0.9	0.6	50	34192	0

0.9	0.6	100	34192	1109
0.9	0.6	300	34192	1571
0.9	0.6	500	34192	1062
0.9	0.7	1	34192	8457
0.9	0.7	3	34192	9506
0.9	0.7	5	34192	8121
0.9	0.7	10	34192	7746
0.9	0.7	50	34192	1571
0.9	0.7	100	34192	394
0.9	0.7	300	34192	1571
0.9	0.7	500	34192	394
0.9	0.8	1	34192	11748
0.9	0.8	3	34192	6246
0.9	0.8	5	34192	7993
0.9	0.8	10	34192	4585
0.9	0.8	50	34192	4817
0.9	0.8	100	34192	2791
0.9	0.8	300	34192	0
0.9	0.8	500	34192	0