



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 № 7 по курсу «Анализ алгоритмов»

Тема Методы решения задачи коммивояжера

Студент Виноградов А. О.

Группа ИУ7-56Б

Оценка (баллы)

Преподаватели Волкова Л. Л., Строганов Ю. В.

Москва — 2023 г.

Содержание

Введение	3
1 Аналитическая часть	4
1.1 Метод на основе муравьиного алгоритма	4
2 Конструкторская часть	7
3 Технологическая часть	12
3.1 Реализация алгоритмов	12
3.2 Функциональные тесты	16
4 Исследовательская часть	18
4.1 Технические характеристики	18
4.2 Демонстрация работы программы	18
4.3 Результаты замеров времени	19
4.4 Постановка эксперимента	20
4.4.1 Класс данных 1	20
4.4.2 Класс данных 2	23
Заключение	27
Список использованных источников	28
Приложение А	29
Приложение Б	49

Введение

Задача оптимизации неэффективных с точки зрения времени выполнения алгоритмов всегда были важны. Одним из таких алгоритмов является полный перебор для решения задачи поиска оптимальных путей.

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

Задачи данной лабораторной работы:

- 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.

1 Аналитическая часть

В этом разделе будет представлена информация о задаче коммивояжера, способах ее решения — методе полного перебора и методе на основе муравьиного алгоритма, а также выдвинуты требования к разрабатываемому программному продукту.

Задача коммивояжера 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. Согласно варианту задания рассматривается разновидность задачи с незамкнутым маршрутом, т.е. возврат в начальную вершину в конце маршрута производиться не должен.

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

1.1 Метод на основе муравьиного алгоритма

Муравьиный алгоритм [2] — метод решения задачи оптимизации, основанный на принципе поведения колонии муравьев.

Муравьи действуют, руководствуясь органами чувств. Муравьи используют не прямой обмен информацией через окружающую среду, посредством феромона. Каждый муравей оставляет на своем пути феромоны, чтобы другие могли ориентироваться. Наибольшее количество феромона остается на наиболее посещаемом пути, посещаемость же может быть связана с длинами ребер.

У муравья есть несколько чувств:

- 1) обоняние — способность чувствовать феромон.
- 2) память — способность запомнить пройденный маршрут;
- 3) зрение — способность определить привлекательность ребра ($\eta = \frac{1}{D}$, где D — длина ребра);

Вероятность перехода муравья по ребру ij вычисляется по формуле

$$p_{k,ij} = \begin{cases} \frac{\eta_{ij}^\alpha \cdot \tau_{ij}^\beta}{\sum_{q \notin J_k} \eta_{iq}^\alpha \cdot \tau_{iq}^\beta}, & j \notin J_k \\ 0, & j \in J_k \end{cases} \quad (1.1)$$

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

$$\tau_{ij}(t+1) = \tau_{ij}(t) \cdot (1-p) + \Delta\tau_{ij}(t). \quad (1.2)$$

При этом

$$\Delta\tau_{ij}(t) = \sum_{k=1}^N \Delta\tau_{ij}^k(t), \quad (1.3)$$

где

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, & \text{ребро посещено муравьем } k \text{ в текущий день } t, \\ 0, & \text{иначе} \end{cases} \quad (1.4)$$

Поскольку вероятность перехода в заданную точку 1.1 не должна быть равна нулю, необходимо обеспечить неравенство $\tau_{ij}(t)$ нулю посредством введения минимального 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 (1.4).

Вывод

В данном разделе была рассмотрена задача коммивояжёра, 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.

На рисунке 2.1 представлена схема алгоритма полного перебора путей.

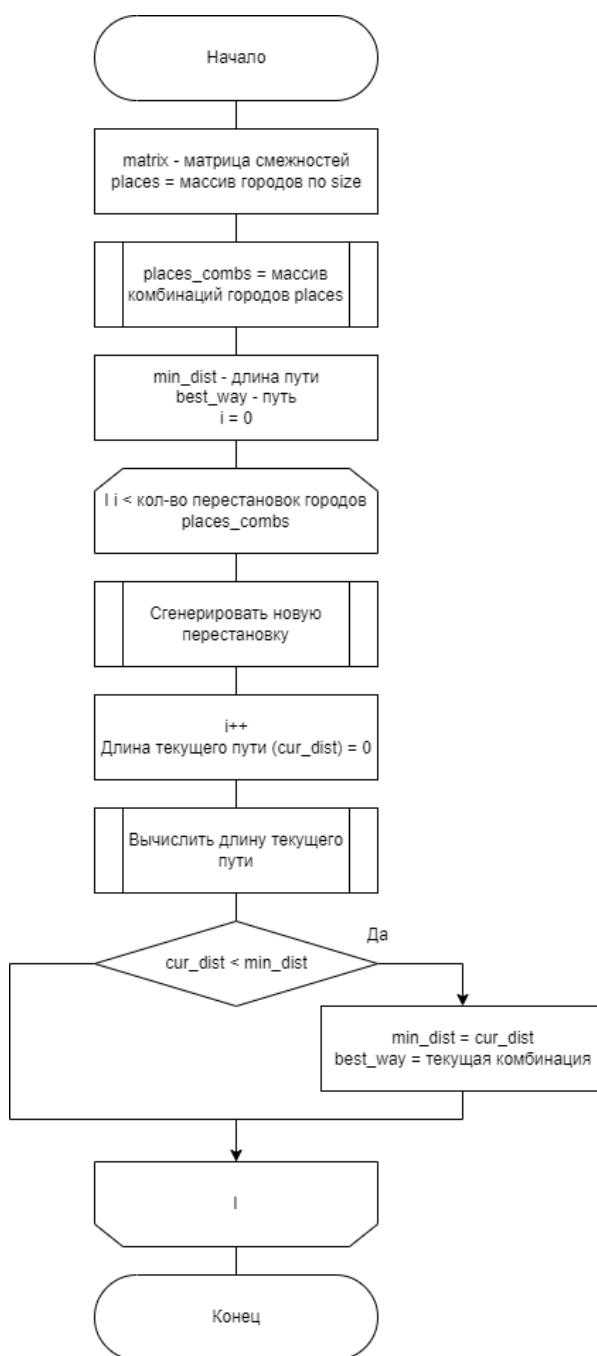


Рисунок 2.1 – Схема алгоритма полного перебора путей

На рисунке 2.2 представлена схема муравьиного алгоритма поиска путей.

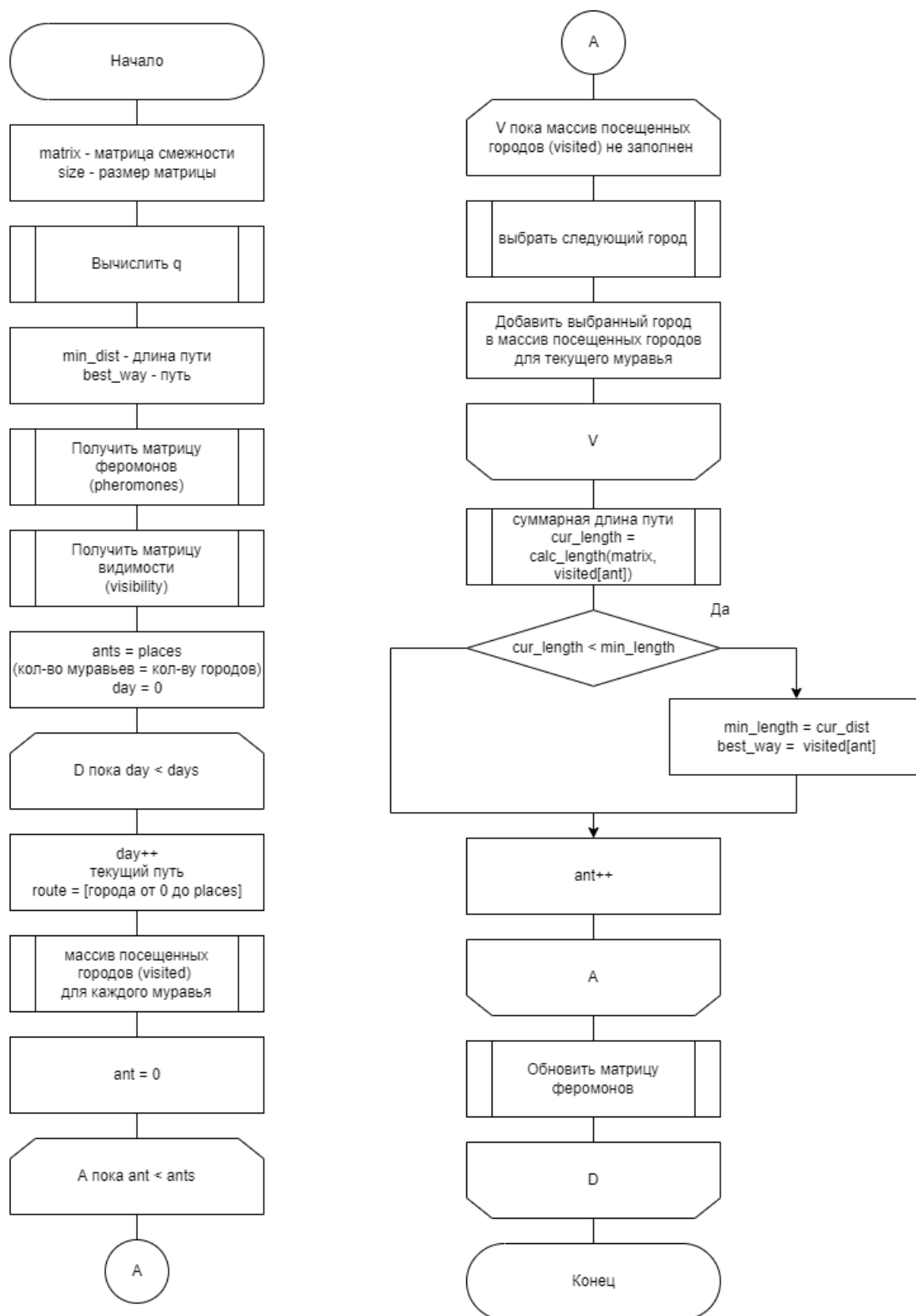


Рисунок 2.2 – Схема муравьиного алгоритма

Также на рисунках 2.3, 2.5, 2.4 представлены схемы вспомогательных функций для муравьиного алгоритма.

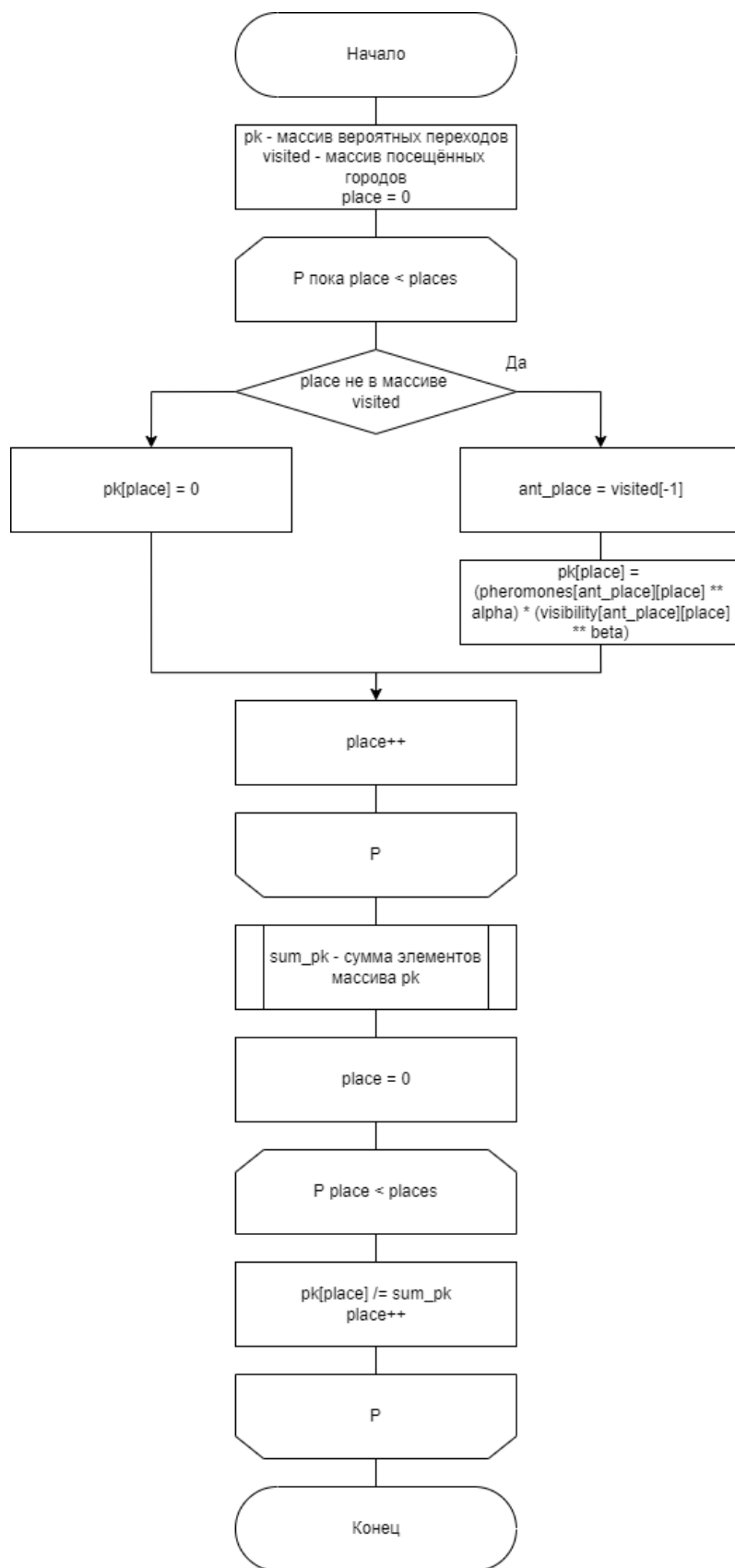


Рисунок 2.3 – Схема алгоритма нахождения массива вероятностей переходов в непосещенные города

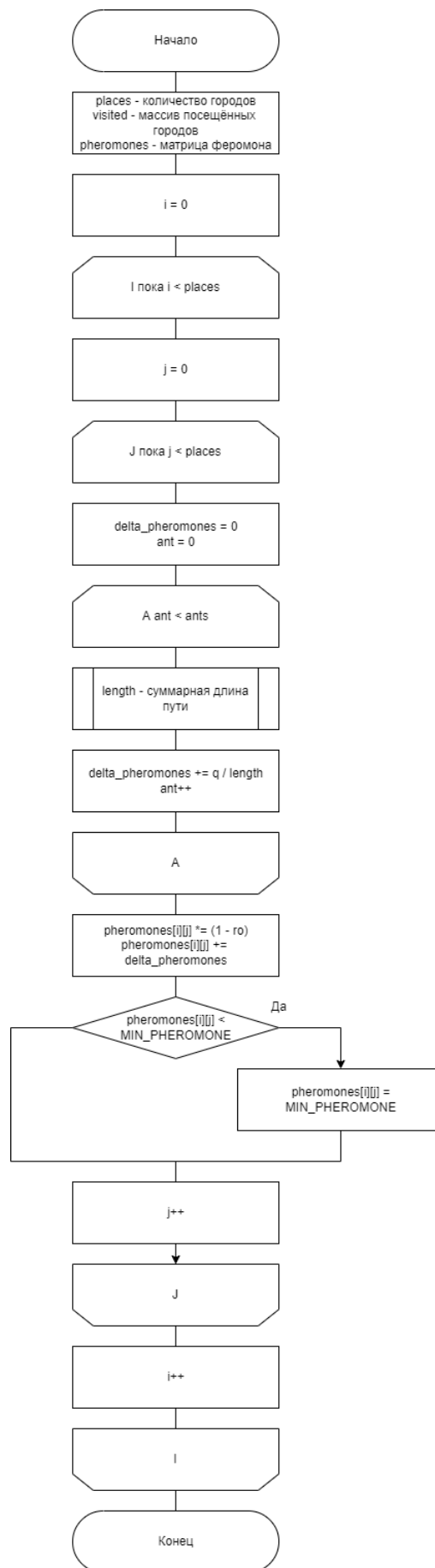


Рисунок 2.4 – Схема алгоритма обновления матрицы феромонов

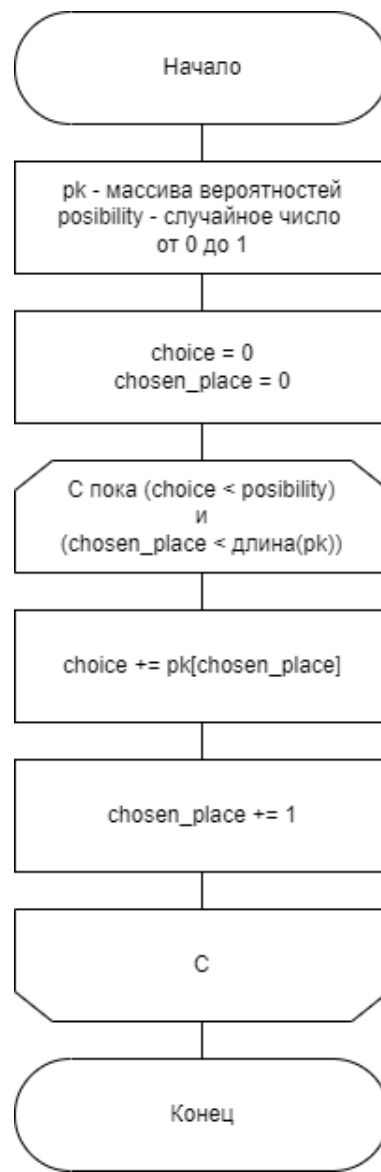


Рисунок 2.5 – Схема алгоритма выбора следующего города

Вывод

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

3 Технологическая часть

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

3.1 Реализация алгоритмов

Для данной работы был выбран язык C++ [3]. Для данной лабораторной работы требуются инструменты для работы с массивами, матрицами, а также замера процессорного времени. Все перечисленные инструменты присутствуют в выбранном языке программирования.

В листингах 3.1, 3.2 представлена реализация алгоритма полного перебора.

Листинг 3.1 – Алгоритм полного перебора (начало)

```
1 pair<double, vector<int>> full_combinations(vector<vector<double>>
    &matrix)
2 {
3     vector<int> places(matrix.size());
4     iota(begin(places), end(places), 0);
5     vector<vector<int>> places_combinations;
6     do {
7         places_combinations.push_back(places);
8     } while (next_permutation(places.begin(), places.end()));
9     vector<int> best_way;
10    double min_dist = numeric_limits<double>::max();
11    for (auto comb : places_combinations)
12    {
13        double cur_dist = 0;
```

Листинг 3.2 – Алгоритм полного перебора (окончание)

```

1      for (int j = 0; j < matrix.size(); ++j)
2      {
3          int start_city = comb[j];
4          int end_city = comb[j + 1];
5          cur_dist += matrix[start_city][end_city];
6      }
7      if (cur_dist < min_dist)
8      {
9          min_dist = cur_dist;
10         best_way = comb;
11     }
12 }
13 return pair<double, vector<int>>(min_dist, best_way);
14 }
```

В листингах 3.3, 3.4 представлена реализация муравьиного алгоритма.

Листинг 3.3 – Муравьиный алгоритм (начало)

```

1 pair<double, vector<int>> ant_alg(vector<vector<double>> &matrix,
2     double alpha, double beta, double ro, int days)
3 {
4     int places = matrix.size();
5     double q = calc_q(matrix, places);
6     vector<int> best_way;
7     double min_dist = numeric_limits<double>::max();
8     vector<vector<double>> pheromones = calc_pheromones(places);
9     vector<vector<double>> visibility = calc_visibility(matrix,
10         places);
11     int ants = places;
12     for (int day = 0; day < days; ++day)
13     {
14         vector<int> route(places);
15         iota(begin(route), end(route), 0);
16         vector<vector<int>> visited = calc_visited_places(route,
17             ants);
18         for (int ant = 0; ant < ants; ++ant)
19         {
```

Листинг 3.4 – Муравьиный алгоритм (окончание)

```

1      while (visited[ant].size() != ants)
2      {
3          vector<double> pk = find_ways(pheromones,
4              visibility, visited, places, ant, alpha, beta);
5          int chosen_place =
6              choose_next_place_by_possibility(pk);
7          visited[ant].push_back(chosen_place - 1);
8      }
9      double cur_length = calc_length(matrix, visited[ant]);
10
11      if (cur_length < min_dist)
12      {
13          min_dist = cur_length;
14          best_way = visited[ant];
15      }
16      pheromones = update_pheromones(matrix, places, visited,
17          pheromones, q, ro);
18  }
19  return pair<double, vector<int>>(min_dist, best_way);
20 }
```

В листингах 3.5 – 3.8 представлены реализации вспомогательных функций для муравьиного алгоритма.

Листинг 3.5 – Алгоритм нахождения массива вероятностей переходов (начало)

```

1  vector<double> find_ways(vector<vector<double>>& pheromones,
2      vector<vector<double>>& visibility, vector<vector<int>>&
3      visited, int places, int ant, double alpha, double beta){
4      vector<double> pk(places, 0);
5      for (int place = 0; place < places; ++place)
6      {
7          if (find(visited[ant].begin(), visited[ant].end(), place)
8              == visited[ant].end())
9          {
10             int ant_place = visited[ant][visited[ant].size() - 1];
11             pk[place] = pow(pheromones[ant_place][place], alpha) * \
12                 pow(visibility[ant_place][place], beta);
13         }
14     }
15 }
```

Листинг 3.6 – Алгоритм нахождения массива вероятностей переходов (окончание)

```
1      else
2          pk[place] = 0;
3      }
4      double sum_pk = summ(pk);
5      for (int place = 0; place < places; ++place)
6          pk[place] /= sum_pk;
7      return pk;
8  }
```

Листинг 3.7 – алгоритм обновления матрицы феромонов

```
1  vector<vector<double>> update_pheromones(vector<vector<double>>&
    matrix, int places, vector<vector<int>>& visited,
    vector<vector<double>> pheromones, double q, double ro)
2  {
3      int ants = places;
4      for (int i = 0; i < places; ++i)
5          for (int j = 0; j < places; ++j)
6              {
7                  double delta = 0;
8                  for (int ant = 0; ant < ants; ++ant)
9                      {
10                         double len = calc_length(matrix, visited[ant]);
11                         delta += q / len;
12                     }
13                 pheromones[i][j] *= (1 - ro);
14                 pheromones[i][j] += delta;
15                 if (pheromones[i][j] < MIN_PHEROMONE)
16                     pheromones[i][j] = MIN_PHEROMONE;
17             }
18      return pheromones;
19  }
```

Листинг 3.8 – Алгоритм выбора следующего города

```

1 int choose_next_place_by_possibility(vector<double> &pk)
2 {
3     double possibility = (((double)rand() + 1) / (RAND_MAX));
4     double choice = 1e-6;
5     int chosen_place = 0;
6
7     while (choice < possibility && chosen_place < pk.size())
8     {
9         choice += pk[chosen_place];
10        ++chosen_place;
11    }
12    if (chosen_place == 0)
13        cout << 1;
14    return chosen_place;
15 }

```

3.2 Функциональные тесты

Применена методология черного ящика. В таблице 3.1 представлены тесты для алгоритмов нахождения гамильтонова пути. Все тесты были успешно пройдены функцией, реализующей полный перебор. Т.к. муравьиный алгоритм не гарантирует точного нахождения кратчайшего пути, тестов для него заготовлено не было.

Таблица 3.1 – Функциональные тесты

Матрица смежности	Ожидаемый результат	Результат программы
$\begin{pmatrix} 0 & 6 & 3 & 9 & 1 \\ 8 & 0 & 3 & 4 & 7 \\ 8 & 4 & 0 & 4 & 6 \\ 8 & 2 & 5 & 0 & 8 \\ 7 & 2 & 4 & 1 & 0 \end{pmatrix}$	7, [0, 4, 3, 1, 2]	7, [0, 4, 3, 1, 2]
$\begin{pmatrix} 0 & 7 & 10 \\ 10 & 0 & 3 \\ 10 & 7 & 0 \end{pmatrix}$	10, [0, 1, 2]	10, [0, 1, 2]
$\begin{pmatrix} 0 & 15 & 19 & 20 \\ 15 & 0 & 12 & 13 \\ 19 & 12 & 0 & 17 \\ 20 & 13 & 17 & 0 \end{pmatrix}$	44, [0, 1, 2, 3]	44, [0, 1, 2, 3]

Вывод

В данном разделе была приведена информация о выбранных средствах для разработки алгоритмов. Для реализации алгоритмов был выбран язык C++. Были представлены листинги и функциональные тесты для каждого из алгоритмов поиска кратчайшего пути.

4 Исследовательская часть

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

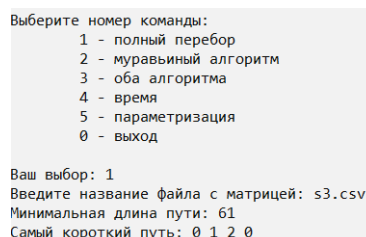
4.1 Технические характеристики

Ниже приведены технические характеристики устройства, на котором было проведено измерение времени работы ПО:

- операционная система Windows 10 Домашняя Версия 21H1 [4] x86_64;
- оперативная память 8 Гбайт 2133 МГц;
- процессор Intel Core i5-8300H с тактовой частотой 2.30 ГГц [5], 4 физических ядра, 8 логических ядер.

4.2 Демонстрация работы программы

На рисунке 4.1 представлен пример результата работы программы. Пользователь, указывая соответствующий пункт меню и название файла с матрицей смежностей, запускает поиск минимального расстояния с помощью алгоритма полного перебора. На экран выводится результат работы алгоритма.



```
Выберите номер команды:
1 - полный перебор
2 - муравьиный алгоритм
3 - оба алгоритма
4 - время
5 - параметризация
0 - выход

Ваш выбор: 1
Введите название файла с матрицей: s3.csv
Минимальная длина пути: 61
Самый короткий путь: 0 1 2 0
```

Рисунок 4.1 – Пример работы программы

4.3 Результаты замеров времени

Время работы было замерено с помощью функции *high_resolution_clock(...)* из заголовочного файла *chrono* на *C++*.

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

Результаты замеров приведены в таблице 4.1, время приведено в миллисекундах.

Таблица 4.1 – Результаты замеров времени (в мс)

Размер матрицы	Полный перебор	Муравьиный алгоритм
2	0.012	3.984
3	0.027	7.487
4	0.103	13.875
5	0.494	24.813
6	3.252	41.180
7	20.901	66.088
8	170.166	96.923
9	1623.428	134.993

На рисунке 4.2 приведена графическая интерпретация результата замеров времени работы реализаций алгоритмов для различных линейных размеров матриц.

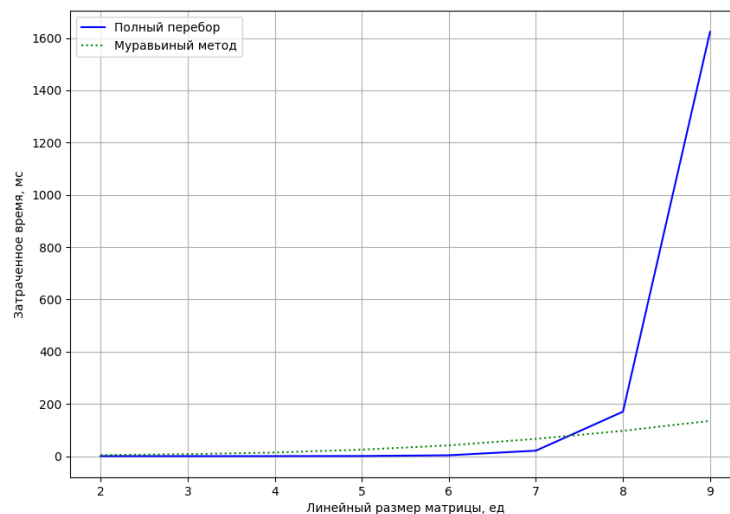


Рисунок 4.2 – Результаты замеров времени работы реализаций алгоритмов для различных линейных размеров матриц

4.4 Постановка эксперимента

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

Итоговая таблица значений параметризации состоит 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;
- delta — разность полученного основаным на муравьином алгоритме методом значения и эталонного значения на данных значениях параметров, показатель качества решения.

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

$$K_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 2 & 1 & 1 & 2 & 1 \\ 1 & 0 & 2 & 2 & 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 0 & 2 & 2 & 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 1 \\ 2 & 1 & 2 & 2 & 0 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 2 & 2 & 0 & 2 & 1 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 0 & 2 & 2 \\ 2 & 1 & 2 & 2 & 2 & 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 0 \end{pmatrix} \quad (4.1)$$

Для данного класса данных приведена таблица 4.4.1 с выборкой параметров, которые наилучшим образом решают поставленную задачу, полные результаты параметризации приведены в приложении А.

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 delta). Если одинаковый результат параметра delta достигается для нескольких кортежей $(\alpha, \rho, \text{days}, \text{result}, \text{delta})$, 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

α	ρ	days	result	delta
0.1	0.1	300	10	0
0.1	0.2	300	10	0
0.1	0.5	300	10	0
0.1	0.6	100	10	0
0.1	0.7	300	10	0
0.2	0.1	500	10	0
0.2	0.2	500	10	1
0.2	0.4	500	10	1
0.2	0.5	500	10	0
0.2	0.7	300	10	0
0.3	0.1	300	10	0
0.3	0.2	300	10	0
0.3	0.4	500	10	0
0.3	0.8	300	10	0
0.3	0.9	300	10	0
0.4	0.1	500	10	0
0.4	0.4	500	10	0
0.4	0.6	500	10	0
0.4	0.7	300	10	0
0.4	0.9	300	10	0
0.5	0.1	500	10	1
0.5	0.3	500	10	0
0.5	0.5	300	10	0
0.5	0.7	300	10	0
0.5	0.9	500	10	1
0.6	0.1	50	10	1
0.6	0.3	500	10	0
0.6	0.5	500	10	1
0.6	0.6	300	10	0
0.6	0.9	500	10	0
0.7	0.2	500	10	0
0.7	0.4	500	10	0
0.7	0.5	300	10	0

0.7	0.7	500	10	0
0.7	0.9	500	10	0
0.8	0.1	500	10	1
0.8	0.3	500	10	1
0.8	0.7	500	10	0
0.8	0.8	500	10	1
0.8	0.9	300	10	0
0.9	0.1	500	10	0
0.9	0.3	500	10	1
0.9	0.5	500	10	0
0.9	0.7	500	10	1
0.9	0.9	500	10	1

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.

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 delta). Если одинаковый результат параметра delta достигается для нескольких кортежей $(\alpha, \rho, \text{days}, \text{result}, \text{delta})$, содержащих одинаковые значения параметров (α, ρ) , среди них выбирается кортеж, содержащий наименьшее значение параметра days.

$$K_2 = \begin{pmatrix} 0 & 3335 & 6874 & 6965 & 6380 & 9302 & 2182 & 2668 & 9645 \\ 3335 & 0 & 2057 & 9364 & 3464 & 4552 & 6097 & 5318 & 4220 \\ 6874 & 2057 & 0 & 2695 & 5333 & 8417 & 2209 & 4219 & 9177 \\ 6965 & 9364 & 2695 & 0 & 1073 & 3715 & 1777 & 6458 & 1082 \\ 6380 & 3464 & 5333 & 1073 & 0 & 3111 & 3677 & 5733 & 2078 \\ 9302 & 4552 & 8417 & 3715 & 3111 & 0 & 8884 & 7863 & 3266 \\ 2182 & 6097 & 2209 & 1777 & 3677 & 8884 & 0 & 7885 & 1221 \\ 2668 & 5318 & 4219 & 6458 & 5733 & 7863 & 7885 & 0 & 9604 \\ 9645 & 4220 & 9177 & 1082 & 2078 & 3266 & 1221 & 9604 & 0 \end{pmatrix} \quad (4.2)$$

Для данного класса данных приведена таблица 4.4.2 с выборкой параметров, которые наилучшим образом решают поставленную задачу, полные результаты параметризации приведены в приложении Б.

Таблица 4.3 – Выборка из параметров для класса данных 2

α	ρ	days	result	delta
0.1	0.1	100	22165	0
0.1	0.3	100	22165	0
0.1	0.6	100	22165	0
0.2	0.2	300	22165	0
0.2	0.6	300	22165	0
0.2	0.9	100	22165	0
0.3	0.3	300	22165	0
0.3	0.6	300	22165	0
0.3	0.8	100	22165	0
0.4	0.2	100	22165	0
0.4	0.3	300	22165	0
0.4	0.5	300	22165	0
0.5	0.2	300	22165	0
0.5	0.4	300	22165	0
0.5	0.8	300	22165	0
0.6	0.1	500	22165	0
0.6	0.3	500	22165	0

0.6	0.8	500	22165	0
0.7	0.4	500	22165	0
0.7	0.7	500	22165	0
0.7	0.9	500	22165	1426
0.8	0.2	500	22165	0
0.8	0.4	500	22165	0
0.8	0.6	500	22165	0
0.9	0.1	500	22165	1426
0.9	0.3	500	22165	1561
0.9	0.7	500	22165	1561

Вывод

В результате исследования было получено, что использование муравьиного алгоритма наиболее эффективно при больших размерах матриц. Так, при размере матрицы, равном 2, муравьиный алгоритм медленнее алгоритма полного перебора в 332 раза, а при размере матрицы, равном 9, муравьиный алгоритм быстрее алгоритма полного перебора в 12 раз. Из рисунка 4.2 следует, что при размерах матриц больше 7 следует использовать муравьиный алгоритм, но стоит учитывать, что он не гарантирует получения глобального минимума при решении задачи.

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

- $\alpha = 0.1, \rho = 0.1, 0.3, 0.6$;
- $\alpha = 0.2, \rho = 0.9$;
- $\alpha = 0.3, \rho = 0.8, 0.9$;
- $\alpha = 0.4, \rho = 0.7$;
- $\alpha = 0.6, \rho = 0.1$.

Следовательно, для класса данных 1 рекомендуется использовать данные параметры.

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

- $\alpha = 0.1, \rho = 0.3, 0.3, 0.7$;
- $\alpha = 0.2, \rho = 0.9$;
- $\alpha = 0.3, \rho = 0.8$;
- $\alpha = 0.4, \rho = 0.5$;
- $\alpha = 0.5, \rho = 0.2, 0.4, 0.8$.

Для второго класса данных 2 рекомендуется использовать данные параметры.

Также во время исследования было замечено, что число дней жизни колонии значительно влияет на качество решения: чем значение параметра *days* больше, тем меньше отклонение решения от эталонного.

Заключение

Цель, которая была поставлена в начале лабораторной работы, была достигнута: была проведена параметризация метода решения задачи коммивояжера на основе муравьиного алгоритма.

Кроме того были выполнены все поставленные задачи:

- 1) была описана задача коммивояжера;
- 2) были описаны методы решения задачи коммивояжера — метод полного перебора и метод на основе муравьиного алгоритма;
- 3) были приведены схемы муравьиного алгоритма и алгоритма, позволяющего решить задачу коммивояжера методом полного перебора;
- 4) был разработан и реализован программный продукт, позволяющий решить задачу коммивояжера исследуемыми методами;
- 5) было произведено сравнение по времени метод полного перебора и метод на основе муравьиного алгоритма;
- 6) был составлен отчет о проделанной работе.

Список использованных источников

- [1] О. Борознов В. Исследование решения задачи коммивояжера. — АГТУ, Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика [Электронный ресурс], 2022. URL: Режим доступа: <https://cyberleninka.ru/article/n/issledovanie-resheniya-zadachi-kommivoyazhera/viewer> (дата обращения: 18.12.2022).
- [2] Семёнов С. С. Педан А. В. Воловиков В. С. Климов И. С. Анализ трудоёмкости различных алгоритмических подходов для решения задачи коммивояжёра. — ООО «Корпорация «Интел Групп», Системы управления, связи и безопасности [Электронный ресурс], 2022. URL: Режим доступа: <https://cyberleninka.ru/article/n/analiz-trudoemkosti-razlichnyh-algoritmicheskikh-podhodov-dlya-resheniya-zadachi-kommivoyazhera> (дата обращения: 18.12.2022).
- [3] C++ language [Электронный ресурс]. Режим доступа: <https://en.cppreference.com/w/cpp/language> (дата обращения: 10.11.2022).
- [4] Windows [Электронный ресурс]. Режим доступа: <https://www.microsoft.com/en-us/windows> (дата обращения: 01.10.2022).
- [5] Процессор Intel Core i5 [Электронный ресурс]. Режим доступа: <https://www.intel.com/processors/core/i5/docs> (дата обращения: 01.10.2022).

Приложение А

Таблица 4.4 – Параметризация для класса данных 1, days — количество дней, result — результат работы, delta — ошибочность полученного результата

α	ρ	days	result	delta
0.1	0.1	1	10	2
0.1	0.1	3	10	2
0.1	0.1	5	10	1
0.1	0.1	10	10	1
0.1	0.1	50	10	1
0.1	0.1	100	10	1
0.1	0.1	300	10	0
0.1	0.1	500	10	0
0.1	0.2	1	10	2
0.1	0.2	3	10	2
0.1	0.2	5	10	2
0.1	0.2	10	10	1
0.1	0.2	50	10	1
0.1	0.2	100	10	1
0.1	0.2	300	10	0
0.1	0.2	500	10	0
0.1	0.3	1	10	3
0.1	0.3	3	10	2
0.1	0.3	5	10	1
0.1	0.3	10	10	1
0.1	0.3	50	10	1
0.1	0.3	100	10	1
0.1	0.3	300	10	0
0.1	0.3	500	10	0

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

0.1	0.8	5	10	1
0.1	0.8	10	10	1
0.1	0.8	50	10	1
0.1	0.8	100	10	1
0.1	0.8	300	10	0
0.1	0.8	500	10	0
0.1	0.9	1	10	2
0.1	0.9	3	10	2
0.1	0.9	5	10	2
0.1	0.9	10	10	1
0.1	0.9	50	10	1
0.1	0.9	100	10	1
0.1	0.9	300	10	0
0.1	0.9	500	10	0
0.2	0.1	1	10	3
0.2	0.1	3	10	2
0.2	0.1	5	10	2
0.2	0.1	10	10	1
0.2	0.1	50	10	1
0.2	0.1	100	10	1
0.2	0.1	300	10	1
0.2	0.1	500	10	0
0.2	0.2	1	10	3
0.2	0.2	3	10	2
0.2	0.2	5	10	2
0.2	0.2	10	10	1
0.2	0.2	50	10	1
0.2	0.2	100	10	1
0.2	0.2	300	10	1
0.2	0.2	500	10	1
0.2	0.3	1	10	3
0.2	0.3	3	10	2
0.2	0.3	5	10	2
0.2	0.3	10	10	2

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

0.2	0.7	300	10	0
0.2	0.7	500	10	0
0.2	0.8	1	10	2
0.2	0.8	3	10	2
0.2	0.8	5	10	2
0.2	0.8	10	10	1
0.2	0.8	50	10	1
0.2	0.8	100	10	1
0.2	0.8	300	10	0
0.2	0.8	500	10	0
0.2	0.9	1	10	2
0.2	0.9	3	10	2
0.2	0.9	5	10	2
0.2	0.9	10	10	1
0.2	0.9	50	10	1
0.2	0.9	100	10	1
0.2	0.9	300	10	0
0.2	0.9	500	10	0
0.3	0.1	1	10	2
0.3	0.1	3	10	2
0.3	0.1	5	10	2
0.3	0.1	10	10	2
0.3	0.1	50	10	1
0.3	0.1	100	10	1
0.3	0.1	300	10	0
0.3	0.1	500	10	0
0.3	0.2	1	10	2
0.3	0.2	3	10	2
0.3	0.2	5	10	1
0.3	0.2	10	10	1
0.3	0.2	50	10	1
0.3	0.2	100	10	1
0.3	0.2	300	10	0
0.3	0.2	500	10	0

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

0.3	0.7	5	10	1
0.3	0.7	10	10	2
0.3	0.7	50	10	1
0.3	0.7	100	10	1
0.3	0.7	300	10	1
0.3	0.7	500	10	0
0.3	0.8	1	10	2
0.3	0.8	3	10	2
0.3	0.8	5	10	2
0.3	0.8	10	10	2
0.3	0.8	50	10	1
0.3	0.8	100	10	1
0.3	0.8	300	10	0
0.3	0.8	500	10	0
0.3	0.9	1	10	2
0.3	0.9	3	10	2
0.3	0.9	5	10	2
0.3	0.9	10	10	2
0.3	0.9	50	10	1
0.3	0.9	100	10	1
0.3	0.9	300	10	0
0.3	0.9	500	10	0
0.4	0.1	1	10	2
0.4	0.1	3	10	2
0.4	0.1	5	10	1
0.4	0.1	10	10	1
0.4	0.1	50	10	1
0.4	0.1	100	10	1
0.4	0.1	300	10	1
0.4	0.1	500	10	0
0.4	0.2	1	10	3
0.4	0.2	3	10	1
0.4	0.2	5	10	2
0.4	0.2	10	10	1

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

0.4	0.6	300	10	1
0.4	0.6	500	10	0
0.4	0.7	1	10	3
0.4	0.7	3	10	2
0.4	0.7	5	10	2
0.4	0.7	10	10	2
0.4	0.7	50	10	1
0.4	0.7	100	10	1
0.4	0.7	300	10	0
0.4	0.7	500	10	0
0.4	0.8	1	10	3
0.4	0.8	3	10	2
0.4	0.8	5	10	2
0.4	0.8	10	10	1
0.4	0.8	50	10	1
0.4	0.8	100	10	1
0.4	0.8	300	10	0
0.4	0.8	500	10	1
0.4	0.9	1	10	3
0.4	0.9	3	10	2
0.4	0.9	5	10	2
0.4	0.9	10	10	2
0.4	0.9	50	10	1
0.4	0.9	100	10	1
0.4	0.9	300	10	0
0.4	0.9	500	10	1
0.5	0.1	1	10	3
0.5	0.1	3	10	2
0.5	0.1	5	10	2
0.5	0.1	10	10	2
0.5	0.1	50	10	1
0.5	0.1	100	10	1
0.5	0.1	300	10	1
0.5	0.1	500	10	1

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

0.5	0.6	5	10	2
0.5	0.6	10	10	2
0.5	0.6	50	10	1
0.5	0.6	100	10	1
0.5	0.6	300	10	1
0.5	0.6	500	10	1
0.5	0.7	1	10	3
0.5	0.7	3	10	2
0.5	0.7	5	10	1
0.5	0.7	10	10	1
0.5	0.7	50	10	1
0.5	0.7	100	10	1
0.5	0.7	300	10	0
0.5	0.7	500	10	1
0.5	0.8	1	10	4
0.5	0.8	3	10	2
0.5	0.8	5	10	1
0.5	0.8	10	10	2
0.5	0.8	50	10	1
0.5	0.8	100	10	1
0.5	0.8	300	10	1
0.5	0.8	500	10	1
0.5	0.9	1	10	2
0.5	0.9	3	10	1
0.5	0.9	5	10	2
0.5	0.9	10	10	2
0.5	0.9	50	10	1
0.5	0.9	100	10	1
0.5	0.9	300	10	1
0.5	0.9	500	10	1
0.6	0.1	1	10	2
0.6	0.1	3	10	1
0.6	0.1	5	10	2
0.6	0.1	10	10	1

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

0.6	0.5	300	10	1
0.6	0.5	500	10	1
0.6	0.6	1	10	2
0.6	0.6	3	10	2
0.6	0.6	5	10	2
0.6	0.6	10	10	2
0.6	0.6	50	10	1
0.6	0.6	100	10	1
0.6	0.6	300	10	0
0.6	0.6	500	10	1
0.6	0.7	1	10	2
0.6	0.7	3	10	2
0.6	0.7	5	10	2
0.6	0.7	10	10	2
0.6	0.7	50	10	1
0.6	0.7	100	10	1
0.6	0.7	300	10	1
0.6	0.7	500	10	0
0.6	0.8	1	10	2
0.6	0.8	3	10	2
0.6	0.8	5	10	2
0.6	0.8	10	10	2
0.6	0.8	50	10	1
0.6	0.8	100	10	1
0.6	0.8	300	10	1
0.6	0.8	500	10	1
0.6	0.9	1	10	3
0.6	0.9	3	10	2
0.6	0.9	5	10	1
0.6	0.9	10	10	2
0.6	0.9	50	10	1
0.6	0.9	100	10	1
0.6	0.9	300	10	1
0.6	0.9	500	10	0

0.7	0.1	1	10	2
0.7	0.1	3	10	2
0.7	0.1	5	10	2
0.7	0.1	10	10	2
0.7	0.1	50	10	1
0.7	0.1	100	10	1
0.7	0.1	300	10	1
0.7	0.1	500	10	1
0.7	0.2	1	10	3
0.7	0.2	3	10	2
0.7	0.2	5	10	2
0.7	0.2	10	10	1
0.7	0.2	50	10	1
0.7	0.2	100	10	1
0.7	0.2	300	10	1
0.7	0.2	500	10	0
0.7	0.3	1	10	3
0.7	0.3	3	10	2
0.7	0.3	5	10	2
0.7	0.3	10	10	2
0.7	0.3	50	10	1
0.7	0.3	100	10	1
0.7	0.3	300	10	1
0.7	0.3	500	10	1
0.7	0.4	1	10	3
0.7	0.4	3	10	2
0.7	0.4	5	10	2
0.7	0.4	10	10	2
0.7	0.4	50	10	1
0.7	0.4	100	10	1
0.7	0.4	300	10	1
0.7	0.4	500	10	0
0.7	0.5	1	10	3
0.7	0.5	3	10	2

0.7	0.5	5	10	2
0.7	0.5	10	10	2
0.7	0.5	50	10	1
0.7	0.5	100	10	1
0.7	0.5	300	10	0
0.7	0.5	500	10	0
0.7	0.6	1	10	3
0.7	0.6	3	10	2
0.7	0.6	5	10	2
0.7	0.6	10	10	2
0.7	0.6	50	10	1
0.7	0.6	100	10	1
0.7	0.6	300	10	1
0.7	0.6	500	10	1
0.7	0.7	1	10	3
0.7	0.7	3	10	2
0.7	0.7	5	10	2
0.7	0.7	10	10	2
0.7	0.7	50	10	1
0.7	0.7	100	10	1
0.7	0.7	300	10	1
0.7	0.7	500	10	0
0.7	0.8	1	10	4
0.7	0.8	3	10	2
0.7	0.8	5	10	2
0.7	0.8	10	10	2
0.7	0.8	50	10	1
0.7	0.8	100	10	1
0.7	0.8	300	10	1
0.7	0.8	500	10	1
0.7	0.9	1	10	3
0.7	0.9	3	10	2
0.7	0.9	5	10	2
0.7	0.9	10	10	2

0.7	0.9	50	10	1
0.7	0.9	100	10	1
0.7	0.9	300	10	1
0.7	0.9	500	10	0
0.8	0.1	1	10	3
0.8	0.1	3	10	2
0.8	0.1	5	10	2
0.8	0.1	10	10	2
0.8	0.1	50	10	1
0.8	0.1	100	10	1
0.8	0.1	300	10	1
0.8	0.1	500	10	1
0.8	0.2	1	10	3
0.8	0.2	3	10	2
0.8	0.2	5	10	2
0.8	0.2	10	10	2
0.8	0.2	50	10	1
0.8	0.2	100	10	1
0.8	0.2	300	10	1
0.8	0.2	500	10	1
0.8	0.3	1	10	3
0.8	0.3	3	10	2
0.8	0.3	5	10	2
0.8	0.3	10	10	2
0.8	0.3	50	10	1
0.8	0.3	100	10	1
0.8	0.3	300	10	1
0.8	0.3	500	10	1
0.8	0.4	1	10	2
0.8	0.4	3	10	3
0.8	0.4	5	10	2
0.8	0.4	10	10	2
0.8	0.4	50	10	1
0.8	0.4	100	10	1

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

0.8	0.9	1	10	3
0.8	0.9	3	10	2
0.8	0.9	5	10	2
0.8	0.9	10	10	1
0.8	0.9	50	10	1
0.8	0.9	100	10	1
0.8	0.9	300	10	0
0.8	0.9	500	10	0
0.9	0.1	1	10	3
0.9	0.1	3	10	2
0.9	0.1	5	10	2
0.9	0.1	10	10	2
0.9	0.1	50	10	1
0.9	0.1	100	10	1
0.9	0.1	300	10	1
0.9	0.1	500	10	0
0.9	0.2	1	10	3
0.9	0.2	3	10	2
0.9	0.2	5	10	2
0.9	0.2	10	10	2
0.9	0.2	50	10	1
0.9	0.2	100	10	1
0.9	0.2	300	10	1
0.9	0.2	500	10	1
0.9	0.3	1	10	2
0.9	0.3	3	10	2
0.9	0.3	5	10	2
0.9	0.3	10	10	2
0.9	0.3	50	10	1
0.9	0.3	100	10	1
0.9	0.3	300	10	1
0.9	0.3	500	10	1
0.9	0.4	1	10	2
0.9	0.4	3	10	2

0.9	0.4	5	10	2
0.9	0.4	10	10	2
0.9	0.4	50	10	1
0.9	0.4	100	10	1
0.9	0.4	300	10	1
0.9	0.4	500	10	1
0.9	0.5	1	10	3
0.9	0.5	3	10	2
0.9	0.5	5	10	2
0.9	0.5	10	10	2
0.9	0.5	50	10	1
0.9	0.5	100	10	1
0.9	0.5	300	10	1
0.9	0.5	500	10	0
0.9	0.6	1	10	3
0.9	0.6	3	10	2
0.9	0.6	5	10	2
0.9	0.6	10	10	1
0.9	0.6	50	10	1
0.9	0.6	100	10	1
0.9	0.6	300	10	1
0.9	0.6	500	10	1
0.9	0.7	1	10	3
0.9	0.7	3	10	2
0.9	0.7	5	10	2
0.9	0.7	10	10	2
0.9	0.7	50	10	1
0.9	0.7	100	10	1
0.9	0.7	300	10	1
0.9	0.7	500	10	1
0.9	0.8	1	10	3
0.9	0.8	3	10	2
0.9	0.8	5	10	2
0.9	0.8	10	10	2

0.9	0.8	50	10	1
0.9	0.8	100	10	1
0.9	0.8	300	10	1
0.9	0.8	500	10	1
0.9	0.9	1	10	4
0.9	0.9	3	10	2
0.9	0.9	5	10	2
0.9	0.9	10	10	2
0.9	0.9	50	10	1
0.9	0.9	100	10	1
0.9	0.9	300	10	1
0.9	0.9	500	10	1

Приложение Б

Таблица 4.5 – Параметризация для класса данных 2, days — количество дней, result — результат работы, delta — ошибочность полученного результата

α	ρ	days	result	delta
0.1	0.1	1	22165	9664
0.1	0.1	3	22165	5437
0.1	0.1	5	22165	4224
0.1	0.1	10	22165	2454
0.1	0.1	50	22165	1561
0.1	0.1	100	22165	0
0.1	0.1	300	22165	0
0.1	0.1	500	22165	0
0.1	0.2	1	22165	9256
0.1	0.2	3	22165	7483
0.1	0.2	5	22165	3136
0.1	0.2	10	22165	4054
0.1	0.2	50	22165	1426
0.1	0.2	100	22165	1305
0.1	0.2	300	22165	0
0.1	0.2	500	22165	0
0.1	0.3	1	22165	11430
0.1	0.3	3	22165	5997
0.1	0.3	5	22165	6726
0.1	0.3	10	22165	2657
0.1	0.3	50	22165	1426
0.1	0.3	100	22165	0
0.1	0.3	300	22165	0
0.1	0.3	500	22165	0

0.1	0.4	1	22165	12147
0.1	0.4	3	22165	5778
0.1	0.4	5	22165	6485
0.1	0.4	10	22165	1707
0.1	0.4	50	22165	1305
0.1	0.4	100	22165	1305
0.1	0.4	300	22165	0
0.1	0.4	500	22165	0
0.1	0.5	1	22165	7611
0.1	0.5	3	22165	5520
0.1	0.5	5	22165	4161
0.1	0.5	10	22165	3136
0.1	0.5	50	22165	1426
0.1	0.5	100	22165	1426
0.1	0.5	300	22165	0
0.1	0.5	500	22165	0
0.1	0.6	1	22165	8125
0.1	0.6	3	22165	8895
0.1	0.6	5	22165	5146
0.1	0.6	10	22165	2454
0.1	0.6	50	22165	1561
0.1	0.6	100	22165	0
0.1	0.6	300	22165	0
0.1	0.6	500	22165	0
0.1	0.7	1	22165	10298
0.1	0.7	3	22165	8125
0.1	0.7	5	22165	3282
0.1	0.7	10	22165	4255
0.1	0.7	50	22165	1561
0.1	0.7	100	22165	0
0.1	0.7	300	22165	0
0.1	0.7	500	22165	0
0.1	0.8	1	22165	9570
0.1	0.8	3	22165	6974

0.1	0.8	5	22165	4300
0.1	0.8	10	22165	2611
0.1	0.8	50	22165	1554
0.1	0.8	100	22165	1561
0.1	0.8	300	22165	0
0.1	0.8	500	22165	0
0.1	0.9	1	22165	9121
0.1	0.9	3	22165	9033
0.1	0.9	5	22165	3876
0.1	0.9	10	22165	3005
0.1	0.9	50	22165	1426
0.1	0.9	100	22165	2454
0.1	0.9	300	22165	0
0.1	0.9	500	22165	0
0.2	0.1	1	22165	7819
0.2	0.1	3	22165	8139
0.2	0.1	5	22165	6262
0.2	0.1	10	22165	2966
0.2	0.1	50	22165	1305
0.2	0.1	100	22165	1426
0.2	0.1	300	22165	0
0.2	0.1	500	22165	0
0.2	0.2	1	22165	9841
0.2	0.2	3	22165	6564
0.2	0.2	5	22165	5153
0.2	0.2	10	22165	2966
0.2	0.2	50	22165	1661
0.2	0.2	100	22165	1426
0.2	0.2	300	22165	0
0.2	0.2	500	22165	0
0.2	0.3	1	22165	9649
0.2	0.3	3	22165	6483
0.2	0.3	5	22165	4178
0.2	0.3	10	22165	4964

0.2	0.3	50	22165	1707
0.2	0.3	100	22165	1426
0.2	0.3	300	22165	1305
0.2	0.3	500	22165	0
0.2	0.4	1	22165	5932
0.2	0.4	3	22165	7514
0.2	0.4	5	22165	7251
0.2	0.4	10	22165	2966
0.2	0.4	50	22165	1305
0.2	0.4	100	22165	1426
0.2	0.4	300	22165	0
0.2	0.4	500	22165	0
0.2	0.5	1	22165	11035
0.2	0.5	3	22165	7732
0.2	0.5	5	22165	3233
0.2	0.5	10	22165	2966
0.2	0.5	50	22165	1554
0.2	0.5	100	22165	1305
0.2	0.5	300	22165	0
0.2	0.5	500	22165	0
0.2	0.6	1	22165	8893
0.2	0.6	3	22165	3713
0.2	0.6	5	22165	6645
0.2	0.6	10	22165	3005
0.2	0.6	50	22165	1554
0.2	0.6	100	22165	1305
0.2	0.6	300	22165	0
0.2	0.6	500	22165	0
0.2	0.7	1	22165	10749
0.2	0.7	3	22165	7565
0.2	0.7	5	22165	6197
0.2	0.7	10	22165	3175
0.2	0.7	50	22165	1600
0.2	0.7	100	22165	1600

0.2	0.7	300	22165	0
0.2	0.7	500	22165	0
0.2	0.8	1	22165	10772
0.2	0.8	3	22165	5876
0.2	0.8	5	22165	8303
0.2	0.8	10	22165	3862
0.2	0.8	50	22165	2408
0.2	0.8	100	22165	2408
0.2	0.8	300	22165	0
0.2	0.8	500	22165	0
0.2	0.9	1	22165	6271
0.2	0.9	3	22165	4766
0.2	0.9	5	22165	4111
0.2	0.9	10	22165	3012
0.2	0.9	50	22165	1305
0.2	0.9	100	22165	0
0.2	0.9	300	22165	0
0.2	0.9	500	22165	0
0.3	0.1	1	22165	15670
0.3	0.1	3	22165	4015
0.3	0.1	5	22165	5876
0.3	0.1	10	22165	4629
0.3	0.1	50	22165	1700
0.3	0.1	100	22165	1305
0.3	0.1	300	22165	0
0.3	0.1	500	22165	0
0.3	0.2	1	22165	7665
0.3	0.2	3	22165	9433
0.3	0.2	5	22165	8836
0.3	0.2	10	22165	4332
0.3	0.2	50	22165	1661
0.3	0.2	100	22165	1561
0.3	0.2	300	22165	1554
0.3	0.2	500	22165	0

0.3	0.3	1	22165	10018
0.3	0.3	3	22165	5279
0.3	0.3	5	22165	5965
0.3	0.3	10	22165	3282
0.3	0.3	50	22165	1305
0.3	0.3	100	22165	2408
0.3	0.3	300	22165	0
0.3	0.3	500	22165	0
0.3	0.4	1	22165	7161
0.3	0.4	3	22165	8723
0.3	0.4	5	22165	4015
0.3	0.4	10	22165	4255
0.3	0.4	50	22165	1554
0.3	0.4	100	22165	1554
0.3	0.4	300	22165	0
0.3	0.4	500	22165	0
0.3	0.5	1	22165	8501
0.3	0.5	3	22165	7893
0.3	0.5	5	22165	7950
0.3	0.5	10	22165	6528
0.3	0.5	50	22165	1305
0.3	0.5	100	22165	1561
0.3	0.5	300	22165	0
0.3	0.5	500	22165	0
0.3	0.6	1	22165	9933
0.3	0.6	3	22165	6328
0.3	0.6	5	22165	5430
0.3	0.6	10	22165	2454
0.3	0.6	50	22165	1554
0.3	0.6	100	22165	1561
0.3	0.6	300	22165	0
0.3	0.6	500	22165	0
0.3	0.7	1	22165	7433
0.3	0.7	3	22165	9925

0.3	0.7	5	22165	5330
0.3	0.7	10	22165	5279
0.3	0.7	50	22165	1554
0.3	0.7	100	22165	1426
0.3	0.7	300	22165	0
0.3	0.7	500	22165	0
0.3	0.8	1	22165	8492
0.3	0.8	3	22165	7893
0.3	0.8	5	22165	4490
0.3	0.8	10	22165	4065
0.3	0.8	50	22165	1426
0.3	0.8	100	22165	0
0.3	0.8	300	22165	1426
0.3	0.8	500	22165	0
0.3	0.9	1	22165	9174
0.3	0.9	3	22165	6241
0.3	0.9	5	22165	3698
0.3	0.9	10	22165	6229
0.3	0.9	50	22165	1305
0.3	0.9	100	22165	1554
0.3	0.9	300	22165	0
0.3	0.9	500	22165	0
0.4	0.1	1	22165	11075
0.4	0.1	3	22165	6730
0.4	0.1	5	22165	8366
0.4	0.1	10	22165	4687
0.4	0.1	50	22165	2657
0.4	0.1	100	22165	1561
0.4	0.1	300	22165	1554
0.4	0.1	500	22165	0
0.4	0.2	1	22165	11255
0.4	0.2	3	22165	6564
0.4	0.2	5	22165	5709
0.4	0.2	10	22165	4255

0.4	0.2	50	22165	1600
0.4	0.2	100	22165	0
0.4	0.2	300	22165	0
0.4	0.2	500	22165	0
0.4	0.3	1	22165	12366
0.4	0.3	3	22165	9273
0.4	0.3	5	22165	7207
0.4	0.3	10	22165	5486
0.4	0.3	50	22165	1554
0.4	0.3	100	22165	1600
0.4	0.3	300	22165	0
0.4	0.3	500	22165	1305
0.4	0.4	1	22165	8115
0.4	0.4	3	22165	3282
0.4	0.4	5	22165	7007
0.4	0.4	10	22165	3175
0.4	0.4	50	22165	2454
0.4	0.4	100	22165	1426
0.4	0.4	300	22165	1554
0.4	0.4	500	22165	0
0.4	0.5	1	22165	11811
0.4	0.5	3	22165	8636
0.4	0.5	5	22165	6519
0.4	0.5	10	22165	5079
0.4	0.5	50	22165	3012
0.4	0.5	100	22165	1305
0.4	0.5	300	22165	0
0.4	0.5	500	22165	0
0.4	0.6	1	22165	13626
0.4	0.6	3	22165	6766
0.4	0.6	5	22165	6882
0.4	0.6	10	22165	3591
0.4	0.6	50	22165	1561
0.4	0.6	100	22165	2905

0.4	0.6	300	22165	1305
0.4	0.6	500	22165	0
0.4	0.7	1	22165	10408
0.4	0.7	3	22165	9238
0.4	0.7	5	22165	3830
0.4	0.7	10	22165	4912
0.4	0.7	50	22165	3175
0.4	0.7	100	22165	1561
0.4	0.7	300	22165	1305
0.4	0.7	500	22165	1426
0.4	0.8	1	22165	11451
0.4	0.8	3	22165	7604
0.4	0.8	5	22165	4378
0.4	0.8	10	22165	3136
0.4	0.8	50	22165	1426
0.4	0.8	100	22165	1305
0.4	0.8	300	22165	1305
0.4	0.8	500	22165	0
0.4	0.9	1	22165	11226
0.4	0.9	3	22165	8053
0.4	0.9	5	22165	5866
0.4	0.9	10	22165	1707
0.4	0.9	50	22165	1426
0.4	0.9	100	22165	1554
0.4	0.9	300	22165	1305
0.4	0.9	500	22165	0
0.5	0.1	1	22165	17768
0.5	0.1	3	22165	9133
0.5	0.1	5	22165	4673
0.5	0.1	10	22165	4636
0.5	0.1	50	22165	3233
0.5	0.1	100	22165	1661
0.5	0.1	300	22165	1554
0.5	0.1	500	22165	0

0.5	0.2	1	22165	12365
0.5	0.2	3	22165	6713
0.5	0.2	5	22165	5876
0.5	0.2	10	22165	3282
0.5	0.2	50	22165	1700
0.5	0.2	100	22165	1426
0.5	0.2	300	22165	0
0.5	0.2	500	22165	1305
0.5	0.3	1	22165	13606
0.5	0.3	3	22165	8062
0.5	0.3	5	22165	3282
0.5	0.3	10	22165	4597
0.5	0.3	50	22165	1561
0.5	0.3	100	22165	1554
0.5	0.3	300	22165	1554
0.5	0.3	500	22165	1305
0.5	0.4	1	22165	13237
0.5	0.4	3	22165	8303
0.5	0.4	5	22165	8576
0.5	0.4	10	22165	5742
0.5	0.4	50	22165	4015
0.5	0.4	100	22165	1700
0.5	0.4	300	22165	0
0.5	0.4	500	22165	0
0.5	0.5	1	22165	10402
0.5	0.5	3	22165	7611
0.5	0.5	5	22165	5866
0.5	0.5	10	22165	3730
0.5	0.5	50	22165	2859
0.5	0.5	100	22165	3005
0.5	0.5	300	22165	1305
0.5	0.5	500	22165	0
0.5	0.6	1	22165	11320
0.5	0.6	3	22165	9620

0.5	0.6	5	22165	7819
0.5	0.6	10	22165	3112
0.5	0.6	50	22165	2966
0.5	0.6	100	22165	1554
0.5	0.6	300	22165	1426
0.5	0.6	500	22165	0
0.5	0.7	1	22165	8056
0.5	0.7	3	22165	7513
0.5	0.7	5	22165	8973
0.5	0.7	10	22165	6619
0.5	0.7	50	22165	2408
0.5	0.7	100	22165	1561
0.5	0.7	300	22165	1305
0.5	0.7	500	22165	1305
0.5	0.8	1	22165	12656
0.5	0.8	3	22165	7597
0.5	0.8	5	22165	5685
0.5	0.8	10	22165	8705
0.5	0.8	50	22165	2657
0.5	0.8	100	22165	1707
0.5	0.8	300	22165	0
0.5	0.8	500	22165	0
0.5	0.9	1	22165	7819
0.5	0.9	3	22165	6882
0.5	0.9	5	22165	6789
0.5	0.9	10	22165	5876
0.5	0.9	50	22165	2905
0.5	0.9	100	22165	1554
0.5	0.9	300	22165	0
0.5	0.9	500	22165	0
0.6	0.1	1	22165	15518
0.6	0.1	3	22165	11628
0.6	0.1	5	22165	8043
0.6	0.1	10	22165	7467

0.6	0.1	50	22165	2905
0.6	0.1	100	22165	2966
0.6	0.1	300	22165	1305
0.6	0.1	500	22165	0
0.6	0.2	1	22165	11692
0.6	0.2	3	22165	10623
0.6	0.2	5	22165	5866
0.6	0.2	10	22165	4178
0.6	0.2	50	22165	4065
0.6	0.2	100	22165	2611
0.6	0.2	300	22165	1554
0.6	0.2	500	22165	1305
0.6	0.3	1	22165	15565
0.6	0.3	3	22165	9709
0.6	0.3	5	22165	5146
0.6	0.3	10	22165	6589
0.6	0.3	50	22165	3112
0.6	0.3	100	22165	1700
0.6	0.3	300	22165	1707
0.6	0.3	500	22165	0
0.6	0.4	1	22165	12048
0.6	0.4	3	22165	11570
0.6	0.4	5	22165	7804
0.6	0.4	10	22165	7161
0.6	0.4	50	22165	3282
0.6	0.4	100	22165	2454
0.6	0.4	300	22165	1305
0.6	0.4	500	22165	0
0.6	0.5	1	22165	10613
0.6	0.5	3	22165	12326
0.6	0.5	5	22165	9535
0.6	0.5	10	22165	4668
0.6	0.5	50	22165	2966
0.6	0.5	100	22165	1600

0.6	0.5	300	22165	1305
0.6	0.5	500	22165	1426
0.6	0.6	1	22165	8616
0.6	0.6	3	22165	10740
0.6	0.6	5	22165	4957
0.6	0.6	10	22165	4964
0.6	0.6	50	22165	3698
0.6	0.6	100	22165	2966
0.6	0.6	300	22165	1305
0.6	0.6	500	22165	1305
0.6	0.7	1	22165	15641
0.6	0.7	3	22165	6742
0.6	0.7	5	22165	8303
0.6	0.7	10	22165	5686
0.6	0.7	50	22165	2905
0.6	0.7	100	22165	1707
0.6	0.7	300	22165	1426
0.6	0.7	500	22165	1305
0.6	0.8	1	22165	12283
0.6	0.8	3	22165	6262
0.6	0.8	5	22165	8385
0.6	0.8	10	22165	3698
0.6	0.8	50	22165	2866
0.6	0.8	100	22165	2408
0.6	0.8	300	22165	1554
0.6	0.8	500	22165	0
0.6	0.9	1	22165	8301
0.6	0.9	3	22165	8358
0.6	0.9	5	22165	5048
0.6	0.9	10	22165	5866
0.6	0.9	50	22165	4178
0.6	0.9	100	22165	2905
0.6	0.9	300	22165	1600
0.6	0.9	500	22165	0

0.7	0.1	1	22165	10807
0.7	0.1	3	22165	11317
0.7	0.1	5	22165	8487
0.7	0.1	10	22165	8056
0.7	0.1	50	22165	3261
0.7	0.1	100	22165	1426
0.7	0.1	300	22165	1661
0.7	0.1	500	22165	1561
0.7	0.2	1	22165	11646
0.7	0.2	3	22165	11763
0.7	0.2	5	22165	9500
0.7	0.2	10	22165	6041
0.7	0.2	50	22165	3282
0.7	0.2	100	22165	1600
0.7	0.2	300	22165	2454
0.7	0.2	500	22165	1554
0.7	0.3	1	22165	17070
0.7	0.3	3	22165	10044
0.7	0.3	5	22165	7476
0.7	0.3	10	22165	5866
0.7	0.3	50	22165	3713
0.7	0.3	100	22165	2454
0.7	0.3	300	22165	1305
0.7	0.3	500	22165	1305
0.7	0.4	1	22165	17227
0.7	0.4	3	22165	10662
0.7	0.4	5	22165	8695
0.7	0.4	10	22165	3978
0.7	0.4	50	22165	2866
0.7	0.4	100	22165	4566
0.7	0.4	300	22165	1561
0.7	0.4	500	22165	0
0.7	0.5	1	22165	10391
0.7	0.5	3	22165	11075

0.7	0.5	5	22165	4378
0.7	0.5	10	22165	7427
0.7	0.5	50	22165	2657
0.7	0.5	100	22165	2866
0.7	0.5	300	22165	1426
0.7	0.5	500	22165	1426
0.7	0.6	1	22165	16529
0.7	0.6	3	22165	9029
0.7	0.6	5	22165	8990
0.7	0.6	10	22165	8035
0.7	0.6	50	22165	3005
0.7	0.6	100	22165	3073
0.7	0.6	300	22165	1305
0.7	0.6	500	22165	1305
0.7	0.7	1	22165	14117
0.7	0.7	3	22165	11165
0.7	0.7	5	22165	9133
0.7	0.7	10	22165	6041
0.7	0.7	50	22165	2454
0.7	0.7	100	22165	2611
0.7	0.7	300	22165	1561
0.7	0.7	500	22165	0
0.7	0.8	1	22165	13568
0.7	0.8	3	22165	10456
0.7	0.8	5	22165	5167
0.7	0.8	10	22165	3876
0.7	0.8	50	22165	4161
0.7	0.8	100	22165	2657
0.7	0.8	300	22165	1554
0.7	0.8	500	22165	1561
0.7	0.9	1	22165	16447
0.7	0.9	3	22165	6742
0.7	0.9	5	22165	9712
0.7	0.9	10	22165	7497

0.7	0.9	50	22165	3112
0.7	0.9	100	22165	2866
0.7	0.9	300	22165	1554
0.7	0.9	500	22165	1426
0.8	0.1	1	22165	11025
0.8	0.1	3	22165	10295
0.8	0.1	5	22165	10449
0.8	0.1	10	22165	7483
0.8	0.1	50	22165	3830
0.8	0.1	100	22165	2966
0.8	0.1	300	22165	1426
0.8	0.1	500	22165	1600
0.8	0.2	1	22165	16511
0.8	0.2	3	22165	10269
0.8	0.2	5	22165	10149
0.8	0.2	10	22165	7854
0.8	0.2	50	22165	4111
0.8	0.2	100	22165	3591
0.8	0.2	300	22165	1700
0.8	0.2	500	22165	0
0.8	0.3	1	22165	18259
0.8	0.3	3	22165	10006
0.8	0.3	5	22165	5690
0.8	0.3	10	22165	9204
0.8	0.3	50	22165	3699
0.8	0.3	100	22165	2454
0.8	0.3	300	22165	2866
0.8	0.3	500	22165	2611
0.8	0.4	1	22165	19537
0.8	0.4	3	22165	10221
0.8	0.4	5	22165	3730
0.8	0.4	10	22165	4587
0.8	0.4	50	22165	4636
0.8	0.4	100	22165	2859

0.8	0.4	300	22165	1600
0.8	0.4	500	22165	0
0.8	0.5	1	22165	16898
0.8	0.5	3	22165	8167
0.8	0.5	5	22165	9293
0.8	0.5	10	22165	8974
0.8	0.5	50	22165	3282
0.8	0.5	100	22165	3862
0.8	0.5	300	22165	1554
0.8	0.5	500	22165	1426
0.8	0.6	1	22165	15075
0.8	0.6	3	22165	12610
0.8	0.6	5	22165	8184
0.8	0.6	10	22165	6628
0.8	0.6	50	22165	1700
0.8	0.6	100	22165	2657
0.8	0.6	300	22165	2454
0.8	0.6	500	22165	0
0.8	0.7	1	22165	19322
0.8	0.7	3	22165	9181
0.8	0.7	5	22165	8800
0.8	0.7	10	22165	8492
0.8	0.7	50	22165	1554
0.8	0.7	100	22165	1700
0.8	0.7	300	22165	1561
0.8	0.7	500	22165	1554
0.8	0.8	1	22165	17012
0.8	0.8	3	22165	11588
0.8	0.8	5	22165	7497
0.8	0.8	10	22165	3112
0.8	0.8	50	22165	3591
0.8	0.8	100	22165	2966
0.8	0.8	300	22165	1305
0.8	0.8	500	22165	1426

0.8	0.9	1	22165	15541
0.8	0.9	3	22165	10673
0.8	0.9	5	22165	7993
0.8	0.9	10	22165	5520
0.8	0.9	50	22165	3175
0.8	0.9	100	22165	2859
0.8	0.9	300	22165	1305
0.8	0.9	500	22165	1661
0.9	0.1	1	22165	15810
0.9	0.1	3	22165	6726
0.9	0.1	5	22165	8031
0.9	0.1	10	22165	8056
0.9	0.1	50	22165	2454
0.9	0.1	100	22165	4255
0.9	0.1	300	22165	1600
0.9	0.1	500	22165	1426
0.9	0.2	1	22165	15651
0.9	0.2	3	22165	8073
0.9	0.2	5	22165	8659
0.9	0.2	10	22165	7708
0.9	0.2	50	22165	3261
0.9	0.2	100	22165	2880
0.9	0.2	300	22165	2408
0.9	0.2	500	22165	1426
0.9	0.3	1	22165	18339
0.9	0.3	3	22165	8303
0.9	0.3	5	22165	11404
0.9	0.3	10	22165	9181
0.9	0.3	50	22165	3282
0.9	0.3	100	22165	4054
0.9	0.3	300	22165	3175
0.9	0.3	500	22165	1561
0.9	0.4	1	22165	12131
0.9	0.4	3	22165	10531

0.9	0.4	5	22165	11241
0.9	0.4	10	22165	8729
0.9	0.4	50	22165	4566
0.9	0.4	100	22165	2454
0.9	0.4	300	22165	1561
0.9	0.4	500	22165	1561
0.9	0.5	1	22165	14979
0.9	0.5	3	22165	11275
0.9	0.5	5	22165	9474
0.9	0.5	10	22165	7375
0.9	0.5	50	22165	3698
0.9	0.5	100	22165	3862
0.9	0.5	300	22165	1554
0.9	0.5	500	22165	1561
0.9	0.6	1	22165	14238
0.9	0.6	3	22165	13106
0.9	0.6	5	22165	11757
0.9	0.6	10	22165	6533
0.9	0.6	50	22165	2454
0.9	0.6	100	22165	2859
0.9	0.6	300	22165	2880
0.9	0.6	500	22165	1600
0.9	0.7	1	22165	14887
0.9	0.7	3	22165	9847
0.9	0.7	5	22165	11624
0.9	0.7	10	22165	10433
0.9	0.7	50	22165	4640
0.9	0.7	100	22165	2966
0.9	0.7	300	22165	1554
0.9	0.7	500	22165	1561
0.9	0.8	1	22165	16405
0.9	0.8	3	22165	12832
0.9	0.8	5	22165	13489
0.9	0.8	10	22165	6271

0.9	0.8	50	22165	3891
0.9	0.8	100	22165	3730
0.9	0.8	300	22165	1600
0.9	0.8	500	22165	1426
0.9	0.9	1	22165	14703
0.9	0.9	3	22165	10313
0.9	0.9	5	22165	7458
0.9	0.9	10	22165	6974
0.9	0.9	50	22165	3012
0.9	0.9	100	22165	3699
0.9	0.9	300	22165	1305
0.9	0.9	500	22165	1707