**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»

# Отчёт

## по лабораторной работе № 6

**Название:** Муравьиный алгоритм

**Дисциплина:** Анализ алгоритмов

Студент ИУ7-55Б (Группа) _____ (Подпись, дата) М.А. Козлов (И.О. Фамилия)

Преподователь _____ (Подпись, дата) Л.Л. Волкова (И.О. Фамилия)

*Москва, 2020*

# Содержание

# Введение

Муравьиный алгоритм – один из эффективных полиномиальных алгоритмов для нахождения приближённых решений задачи коммивояжёра, а также решения аналогичных задач поиска маршрутов на графах.

Целью данной лабораторной работы является изучение муравьиных алгоритмов и приобретение навыков параметризации методов на примере муравьиного алгоритма, примененного к задаче коммивояжера.

Задачи данной лабораторной работы:

1) рассмотренть муравьиный алгоритм и алгоритм полного перебора в задаче коммивояжера;
2) реализовать эти алгоритмы;
3) сравнить время работы этих алгоритмов.

# 1 Аналитический раздел

В данном разделе будут рассмотрены алгоритмы.

# 2 Констукторский раздел

В данном разделе будут рассмотрены схемы алгоритмов, требования к функциональности ПО, и опредены способы тестирования.

## 2.1 Разработка алгоритмов

Ниже будут представлены схемы алгоритмов:

1) алгоритм 1 (рисунок **??**);
2) алгоритм 2 (рисунок **??**);
3) алгоритм 3 (рисунок **??**).

## 2.2 Требования к функциональности ПО

В данной работе требуется обеспечить следующую минимальную функциональность консольного приложения:

1) ;
2) .

## 2.3 Тестирование

Тестирование ПО будет проводиться методом чёрного ящика. Необходимо проверить работу системы на случаях, когда .

# 3 Технологический раздел

В данном разделе будут выбраны средства реализации ПО и представлен листинг кода.

## 3.1 Средства реализации

В данной работе используется язык программирования python [1], так как он позволяет написать программу в относительно малый срок. В качестве среды разработки использовалась Visual Studio Code [2].

Для замера процессорного времени была использована функция process_time модуля time [3]. Она возвращает значение в долях секунды суммы системного и пользовательского процессорного времени текущего процесса и не включает время, прошедшее во время сна.

## 3.2 Листинг программы

Ниже представлены листинги кода алгоритмов:

1) полным перебором (листинг 3.1);
2) бинарным поиском (листинг 3.2);
3) поиском по сегментам (листинг 3.3).

Листинг 3.1 — Реализация алгоритма поиска слов в словаре полным перебором

Листинг 3.2 — Реализация алгоритма двоичного поиска слова в словаре

Листинг 3.3 — Реализация алгоритма поиска слова в словаре по сегментам

## 3.3 Тестирование

В таблице **??** отображён возможный набор тестов для тестирования методом чёрного ящика, результаты которого, представленные на рисунке 3.1, подтверждают прохождение программы перечисленных тестов.

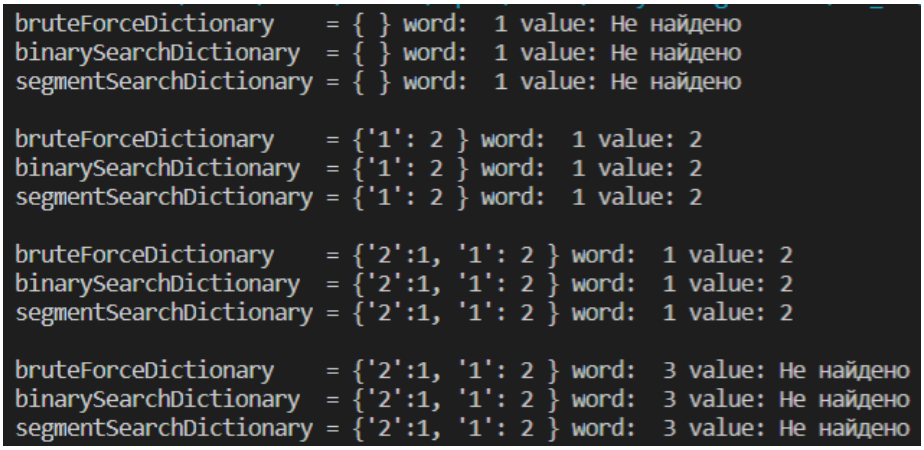


Рисунок 3.1 — Результаты тестирования алгоритмов.

# 4 Экспериментальный раздел

В данном разделе будут проведены эксперименты для проведения сравнительного анализа трёх алгоритмов по затрачиваемому процессорному времени в зависимости от индекса слова в словаре. Тестирование проводилось на ноутбуке с процессором Intel(R) Core(TM) i5-7200U CPU 2.50 GHz [4] под управлением Windows 10 с 8 Гб оперативной памяти.

## 4.1 Сравнительный анализ на основе замеров времени работы алгоритмов

В рамках данного проекта были проведёны эксперименты по замеру времени работы алгоритмов :

1) 1 (график **??**);
2) 2 (график **??**);
3) 3 (график **??**).

## 4.2 Вывод

В ходе экспериментов по замеру времени работы было установлено, что

## Заключение

В ходе выполнения данной лабораторной работы были изучены

## Список использованных источников

1. Python. // [Электронный ресурс]. Режим доступа: https://www.python.org/, (дата обращения: 01.10.2020).

2. Visual Studio Code - Code Editing. // [Электронный ресурс]. Режим доступа: https://code.visualstudio.com, (дата обращения: 01.10.2020).

3. Process time. // [Электронный ресурс]. Режим доступа: https://docs-python.ru/standart-library/modul-time-python/funktsija-process-time-modulja-time, (дата обращения: 01.10.2020).

4. Intel® Core™ i5-7200U Processor. // [Электронный ресурс]. Режим доступа: https://www.intel.com/content/www/us/en/products/processors/core/i5-processors/i5-7200u.html, (дата обращения: 26.09.2020).