

Поздравляю всех, кто читает этот файл, с наступающим/наступившим новым 2018 годом! Желаю успехов, вдохновения, любви и удачи! 🎅🎅🎅

## README – Алгоритм Дамерау-Левенштейна

### 1. *Что делает?*

**Расстояние Дамерау-Левенштейна** — это мера разницы двух строк символов, определяемая как минимальное количество операций вставки, удаления, замены и транспозиции (перестановки двух соседних символов), необходимых для перевода одной строки в другую. Является модификацией расстояния Левенштейна: к операциям вставки, удаления и замены символов, определённых в расстоянии Левенштейна добавлена операция транспозиции (перестановки) символов.

**Подробнее:** [https://ru.wikipedia.org/wiki/Расстояние\\_Дамерау\\_—\\_Левенштейна](https://ru.wikipedia.org/wiki/Расстояние_Дамерау_—_Левенштейна) и [https://neerc.ifmo.ru/wiki/index.php?title=Задача\\_о\\_расстоянии\\_Дамерау-Левенштейна](https://neerc.ifmo.ru/wiki/index.php?title=Задача_о_расстоянии_Дамерау-Левенштейна)

### 2. *Консольная версия программа*

Консольная версия программы определяет расстояние Д.-Л. для двух строк, введенных интерактивно с клавиатуры или загруженных из файла.

#### **Ограничения, наложенные на программу:**

- Стоимость всех операций (удаления, замены, вставки, смены местами двух соседних) неизменна и равна единице.
- Максимально допустимая длина строки для ввода – 100 000 (время работы алгоритма для двух строк по 100 000 будет примерно 1000 с, тестирование на еще больших данных мне не представляется необходимым)

#### **Доступные функции:**

`dldist (строка1) (строка2)`

Рассчитывает расстояние Д.Л. для введенных двух строк.

Пример ввода/вывода:

```
dldist мама папа
Строка 1: мама
Строка 2: папа
Расстояние Дамерау-Левенштейна: 2
Время работы программы: 0ms.
```

Ввод пустой строки не допускается. Строки разделены между собой пробелами. После двух строк дальнейший ввод игнорируется.

---

Загружает содержимое файлов в строки и рассчитывает расстояние Д.Л. для загруженных строк. Файлы для загрузки должны лежать в папке с исполняемым файлом.

[illegible]

Запуск системы на заранее подготовленных тестах, содержащих подводные камни. Сравнение ответа, которых рассчитан алгоритмом с правильным ответом, посчитанным вручную.

```
correcttest
Строка 1: cat
Строка 2: dog
Расстояние Дамерау-Левенштейна: 3
Время работы программы: 0ms.

Правильный ответ: 3

Строка 1: cat
Строка 2: cc
Расстояние Дамерау-Левенштейна: 2
Время работы программы: 0ms.

Правильный ответ: 2

Строка 1: cat
Строка 2:
Расстояние Дамерау-Левенштейна: 3
Время работы программы: 0ms.

Правильный ответ: 3
```

Проверка гипотезы о расходе памяти  $O(s_1 \cdot s_2)$ , где  $s_1, s_2$  – длина соответствующих строк

Для корректной работы (а именно построения по полученным данным реального графика) требуется, чтоб на компьютере была установлена программа MATLAB.

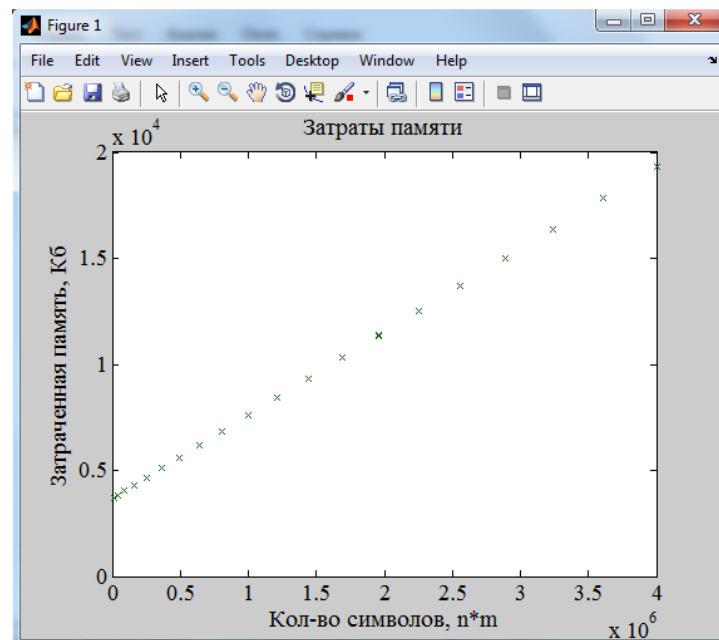
Последовательная прогонка алгоритма на различных строках длиной 100/100, 200/200 ... 2000/2000 с замером выделенной памяти. Дальнейшее построение

графика по полученным данным. Каждый замер проводится трижды для уменьшения погрешности одного измерения.

Пример ввода/вывода:

```
memtest
Исходные затраты памяти : 3676 Кб
Память для <100 x 100> : 3725 Кб
Память для <100 x 100> : 3717 Кб
Память для <100 x 100> : 3717 Кб
Память для <200 x 200> : 3845 Кб
Память для <200 x 200> : 3836 Кб
Память для <200 x 200> : 3836 Кб
Память для <300 x 300> : 4042 Кб
Память для <300 x 300> : 4034 Кб
Память для <300 x 300> : 4034 Кб
Память для <400 x 400> : 4309 Кб
Память для <400 x 400> : 4309 Кб
Память для <400 x 400> : 4309 Кб
```

Пример построенного графика:



---

### speedtest

Проверка гипотезы о временной сложности  $O(s1 \cdot s2)$ , где  $s1$ ,  $s2$  – длина соответствующих строк.

Последовательная прогонка алгоритма на различных строках длиной 100/100, 200/200 ... 2000/2000 с замером времени работы алгоритма. Дальнейшее построение графика по полученным данным. Каждый замер проводится трижды для уменьшения погрешности одного измерения.

Исходя из того, что консольная версия программы написана на C#, не ставилась цель минимизации затраченного времени. Проверялась на практике выведенная теоретическая зависимость.

Для корректной работы (а именно построения по полученным данным реального графика) требуется, чтоб на компьютере была установлена программа MATLAB.

Пример ввода/вывода:

```
speedtest
Время для <100 x 100> : 0 ms
Время для <100 x 100> : 1 ms
Время для <100 x 100> : 1 ms

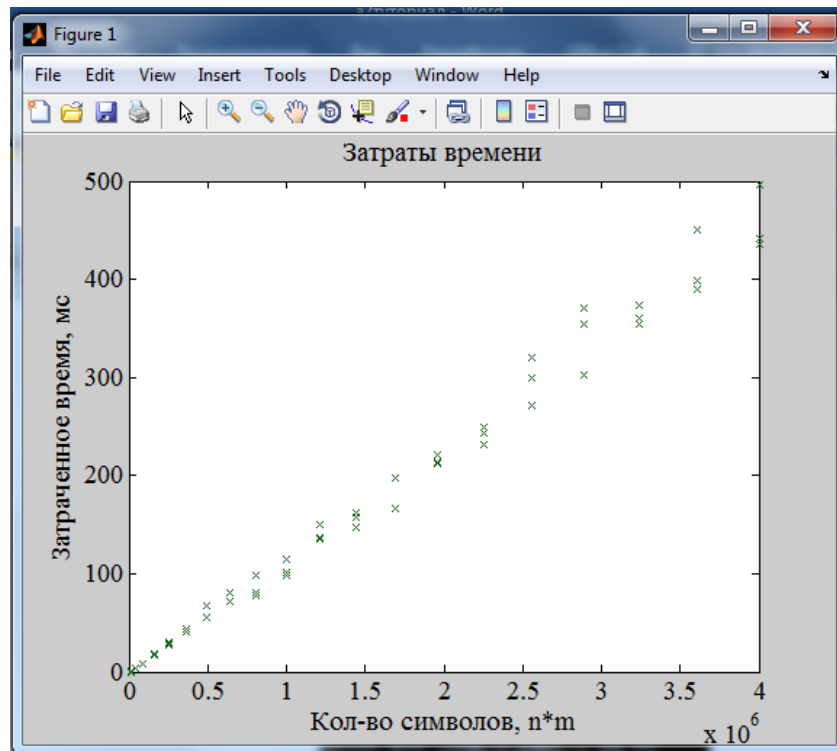
Время для <200 x 200> : 3 ms
Время для <200 x 200> : 3 ms
Время для <200 x 200> : 3 ms

Время для <300 x 300> : 8 ms
Время для <300 x 300> : 8 ms
Время для <300 x 300> : 8 ms

Время для <400 x 400> : 19 ms
Время для <400 x 400> : 19 ms
Время для <400 x 400> : 17 ms

Время для <500 x 500> : 29 ms
Время для <500 x 500> : 28 ms
Время для <500 x 500> : 30 ms
```

Пример построенного графика:



exit

Выход из программы.

Для работы консольной программы требуется наличие папки Tests в корневой директории программы. Файлы из неё загружаются в качестве тестов для измерения времени работы и затрат памяти алгоритма.