

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Н. Э. БАУМАНА

УДК _____

№ госрегистрации _____

Инв. № _____

УТВЕРЖДАЮ

Преподаватель

«_____» _____ 2019 г.

ДИСЦИПЛИНА АНАЛИЗ АЛГОРИТМОВ
ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Расстояние Левенштейна
(промежуточный)

Студент

_____ Ф.М. Набиев

Преподаватели

Л.Л. Волкова, Ю.В. Строганов

Москва, 2019

СОДЕРЖАНИЕ

Введение	3
1 Аналитический раздел	4
1.1 Описание алгоритмов	4
1.1.1 Алгоритм Левенштейна	4
1.1.2 Алгоритм Дамерау-Левенштейна	5
2 Конструкторский раздел	6
2.1 Модель	6
2.2 Разработка алгоритмов	6
2.2.1 Алгоритм Вагнера-Фишера	6
2.2.2 Матричный алгоритм Дамерау-Левенштейна	8
3 Технологический раздел	9
4 Исследовательский раздел	10
Заключение	11

ВВЕДЕНИЕ

Целью данной работы является изучение динамического программирования на материале алгоритмов Левенштейна и Дамерау-Левенштейна.

Данные алгоритмы решают проблему поиска редакционного расстояния между двумя строками. Редакционное расстояние определяется количеством некоторых операций, необходимых для превращения одного слова в другое, а так же стоимостью этих операций.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучение алгоритмов Левенштейна и Дамерау-Левенштейна нахождения расстояния между строками;
- применение метода динамического программирования для матричной реализации указанных алгоритмов;
- получение практических навыков реализации указанных алгоритмов: двух алгоритмов в матричной версии и одного из алгоритмов в рекурсивной версии;
- сравнительный анализ линейной и рекурсивной реализаций выбранного алгоритма определения расстояния между строками по затрачиваемым ресурсам (времени и памяти);
- экспериментальное подтверждение различий во временной эффективности рекурсивной и нерекурсивной реализаций выбранного алгоритма определения расстояния между строками при помощи разработанного программного обеспечения на материале замеров процессорного времени выполнения реализации на варьирующихся длинах строк;
- описание и обоснование полученных результатов в отчете о выполненной лабораторной работе, выполненного как расчётно-пояснительная записка к работе.

1 Аналитический раздел

1.1 Описание алгоритмов

Алгоритм Дамерау-Левенштейна является модификацией алгоритма Левенштейна. Рассмотрим данные методы подробнее.

1.1.1 Алгоритм Левенштейна

Расстояние Левенштейна между двумя строками - это минимальная сумма произведений количества операций вставки, удаления и замены одного символа, необходимых для превращения одной строки в другую, на их стоимость.

Вышеописанные операции имеют следующие обозначения:

- I (insert) - вставка;
- D (delete) - удаление;
- R (replace) - замена;

При этом $\text{cost}(x)$ есть обозначение стоимости некоторой операции x . Будем считать, что символы в строках нумеруются с первого. Пусть S_1 и S_2 - две строки с длинами N и M соответственно. Тогда расстояние Левенштейна вычисляется по формуле $D(M, N)$:

$$D(i,j) = \begin{cases} 0, & i = 0, j = 0 \\ i * \text{cost}(D), & j = 0, i > 0 \\ j * \text{cost}(I), & i = 0, j > 0 \\ D(i-1, j-1), & S_1[i] = S_2[j] \\ \min(\\ D(i, j-1) + \text{cost}(I), \\ D(i-1, j) + \text{cost}(D), & j > 0, i > 0, S_1[i] \neq S_2[j] \\ D(i-1, j-1) + \text{cost}(R) \\), & \end{cases}$$

где $\min(a, b, c)$ возвращает наименьшее значение из a, b, c .

1.1.2 Алгоритм Дамерау-Левенштейна

Определение расстояния Дамерау-Левенштейна аналогично определению расстояния Левенштейна с учётом новой операции - перестановки соседних символов. Соответственно, обозначения операций:

- I (insert) - вставка;
- D (delete) - удаление;
- R (replace) - замена;
- T (transpose) - перестановка соседних символов.

При тех же обозначениях имеем формулу:

$$D(i,j) = \begin{cases} \min(A, D(i-2, j-2) + \text{cost}(T), & i > 1, j > 1, \\ & S_1[i] = S_2[j-1], \\ & S_1[i-1] = S_2[j] \\ A & \text{Иначе} \end{cases}$$

где A:

$$A = \begin{cases} 0, & i = 0, j = 0 \\ i * \text{cost}(D), & j = 0, i > 0 \\ j * \text{cost}(I), & i = 0, j > 0 \\ D(i-1, j-1), & S_1[i] = S_2[j] \\ \min(& \\ D(i, j-1) + \text{cost}(I), & \\ D(i-1, j) + \text{cost}(D), & j > 0, i > 0, S_1[i] \neq S_2[j] \\ D(i-1, j-1) + \text{cost}(R) & \\), & \end{cases}$$

2 Конструкторский раздел

2.1 Модель

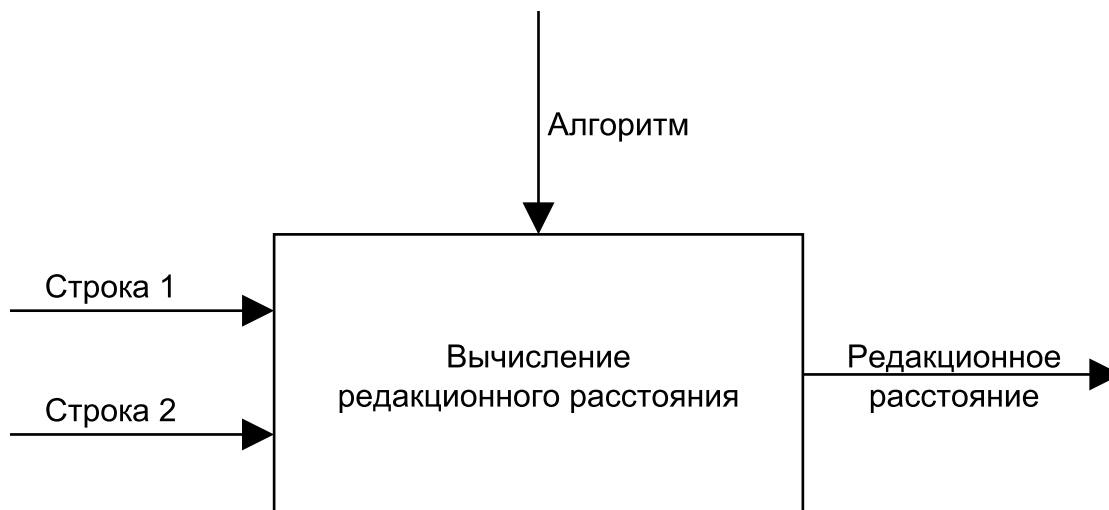


Рисунок 2.1 — IDEF0 модель

2.2 Разработка алгоритмов

2.2.1 Алгоритм Вагнера-Фишера

Алгоритм Вагнера-Фишера является матричной реализацией поиска расстояния Левенштейна. Ниже приведена блок-схема данного алгоритма.

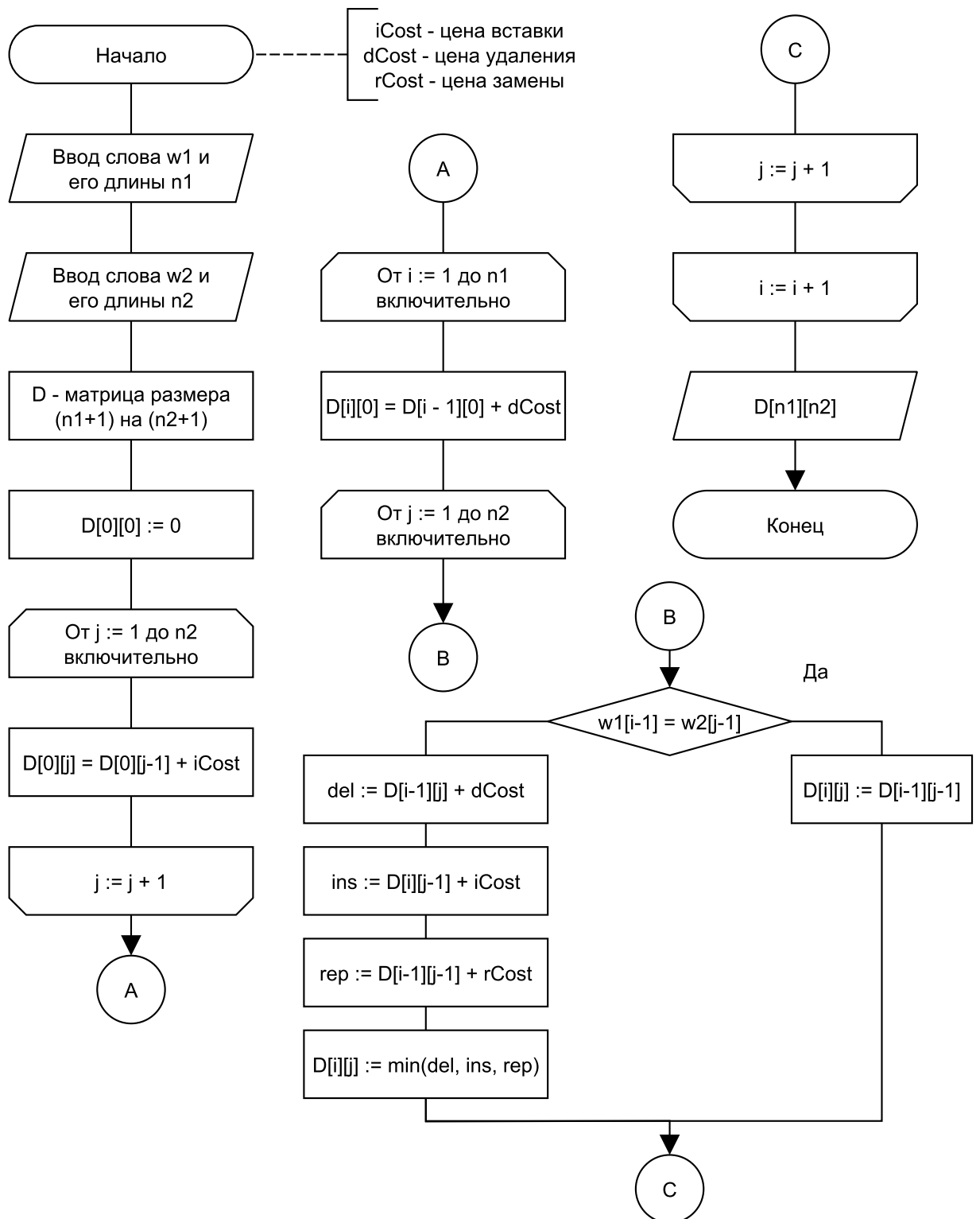


Рисунок 2.2 — Алгоритм Вагнера-Фишера

2.2.2 Матричный алгоритм Дамерау-Левенштейна

3 Технологический раздел

4 Исследовательский раздел

ЗАКЛЮЧЕНИЕ