

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО

Научный руководитель,
Заместитель декана по учебно-
методической работе факультета
компьютерных наук
доцент департамента больших данных и
информационного поиска, канд.
социологических наук

И.Ю. Самоненко

« 13 » _____ мая 2022 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

В.В. Шилов

« ____ » _____ 2022 г.

Приложение для визуализации алгоритма Фараха

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.10.03-01 51 01-1-ЛУ

Исполнитель:

студент группы БПИ206

« 13 » _____ мая 2022 г. / Г. В. Вавилов /

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2022

УТВЕРЖДЕН
RU.17701729.10.03-01 81 01-1-ЛУ

Приложение для визуализации алгоритма Фараха

Программа и методика испытаний

RU.17701729.10.03-01 51 01-1

Листов 14

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Москва 2022

Содержание

1.	Объект испытаний	3
1.1.	Наименование программы.....	3
1.2.	Область применения	3
1.3.	Обозначение испытуемой программы	3
2.	Цель испытаний	4
3.	Требования к программе	5
3.1.	Требования к функциональным характеристикам	5
3.2.	Требования к надежности.....	5
3.3.	Требования к информационной и программной совместимости.	6
4.	Требования к программной документации	7
4.1.	Состав программной документации.....	7
5.	Средства и порядок испытаний	8
5.1.	Технические средства	8
5.2.	Программные средства	8
5.2.1.	Тестовая сборка.....	8
5.2.2.	Средства запуска.....	8
5.3.	Порядок проведения испытаний.....	8
6.	Методы испытаний	9
6.1.	Функциональные требования.....	9
6.2.	Требованиям к информационной и программной совместимости.....	13
6.3.	Требования к надежности.....	13
	Лист регистрации изменений.....	14

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

1. Объект испытаний

1.1. Наименование программы

1.1.1. Название темы разработки на русском языке:

Приложение для визуализации алгоритма Фараха.

1.1.2. Название темы разработки на английском языке:

Farach Algorithm Visualization Application

1.2. Область применения

Алгоритм Фараха используется для построения сжатого суффиксного дерева для строки.

Алгоритм очень объемный и сложный для понимания. Приложение для визуализации должно помочь пользователю с формированием представления о работе алгоритма и об используемых в нем способах хранения данных.

Основная область применения – сфера образования.

1.3. Обозначение испытываемой программы

Приложение для визуализации алгоритма Фараха.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

2. Цель испытаний

Цель проведения испытаний — проверка соответствия характеристик разработанной программы функциональным требованиям, требованиям к информационной и программной совместимости, требованиям к надежности, изложенных в программном документе «Приложение для визуализации алгоритма Фараха» Техническое задание.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

3. Требования к программе

3.1. Требования к функциональным характеристикам

Приложение состоит только из клиентской части, которая должна быть реализована в виде веб-приложения, запускаемого в браузере.

В приложении должно быть реализовано следующее:

1. Поле ввода для строки, по которой будет строиться сжатое суффиксное дерево.
2. Поэтапная визуализация алгоритма Фараха
 - 2.1. Разбиение строки на пары символов и дальнейшее сжатие.
 - 2.1.1. Манипуляции с символами и парами символов показываются при помощи перемещения текстовых элементов.
 - 2.2. Построение четного суффиксного дерева.
 - 2.2.1. Дерево состоит из вершин и ребер. Вершина показывается в виде круга. Ребро показывается в виде отрезка, соединяющего вершины. Ребро подписывается текстом, который оно обозначает.
 - 2.3. Построение нечетного суффиксного дерева.
 - 2.3.1. Требования к дереву аналогичны требованиям в пункте “Построение четного суффиксного дерева”.
 - 2.4. Слияние деревьев.
 - 2.4.1. Требования к дереву аналогичны требованиям в пункте “Построение четного суффиксного дерева”.
 - 2.4.2. Ребра четного дерева имеют цвет, отличный от цвета ребер нечетного дерева.
 - 2.5. Удаление двойных ребер.
3. Возможность приостановить визуализацию.
4. Возможность перейти к следующему кадру визуализации.
5. Возможность перейти к предыдущему кадру визуализации.
6. Возможность построения сжатого суффиксного дерева для новой строки без перезапуска приложения.

3.2. Требования к надежности.

- 3.2.1. Должна быть предусмотрена корректная обработка ошибок, связанных с некорректными действиями пользователя. В случае некорректных входных данных программа не должна завершаться аварийно или уходить в бесконечный цикл. Пользователь должен быть уведомлен о некорректных данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

3.2.2. Во время загрузки программы должно быть обеспечено стабильное соединение с сетью интернет.

3.2.3. После загрузки программа должна иметь возможность корректной работы без соединения с сетью интернет.

3.3. Требования к информационной и программной совместимости.

3.3.1. Требования к информационным структурам на входе.

Входные данные – строка символов, по которой требуется построить суффиксное дерево.

Вводится пользователем в поле ввода на веб-странице.

3.3.2. Требования к информационным структурам на выходе.

В процессе работы программы должен быть визуализирован каждый этап алгоритма.

Результат работы алгоритма – построенное сжатое суффиксное дерево. Данные и структуры данных, использованные во время работы алгоритма, должны быть представлены пользователю в удобном для восприятия виде.

Узлы дерева могут быть представлены в виде окружностей, ребра дерева в виде отрезков, соединяющих вершины.

3.3.3. Требования к исходным кодам и языкам программирования.

Исходный код программы должен быть написан на языках программирования HTML, CSS и TypeScript с возможностью транспилиции в JavaScript.

3.3.4. Требования к программным средствам, используемым программой.

Необходим веб-браузер Firefox > 96.0.0 / Chrome > 97.0.0 или их аналоги на движках Gecko/Blink.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

4. Требования к программной документации

4.1. Состав программной документации

- 4.1.1. «Приложение для визуализации алгоритма Фараха» Техническое задание (ГОСТ 19.201-78)
- 4.1.2. «Приложение для визуализации алгоритма Фараха» Программа и методика испытаний (ГОСТ 19.301-79) [5]
- 4.1.3. Приложение для визуализации алгоритма Фараха» Пояснительная записка (ГОСТ 19.404-79)
- 4.1.4. «Приложение для визуализации алгоритма Фараха» Текст программы (ГОСТ 19.401-78)
- 4.1.5. «Приложение для визуализации алгоритма Фараха» Руководство оператора (ГОСТ 19.505-79)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

5. Средства и порядок испытаний

5.1. Технические средства

Тестирование программы производилось на устройстве со следующими характеристиками:

- Процессор: AMD Ryzen 5 3500U
 - Видеокарта: GeForce GTX 1050.
 - Операционная система: Windows 10
 - Оперативная память: 6ГБ.
 - Доступ в интернет.
 - Клавиатура.
 - Мышь.

5.2. Программные средства

5.2.1. Тестовая сборка

Сборка программы из исходного кода производилась при помощи webpack в режиме “production”. Файл конфигурации [webpack.config.js](#) можно найти в корне проекта.

Работа webpack осуществлялась при помощи установленной среды исполнения Node.js v16.13.2.

Результат сборки содержит 2 файла.

- index.html
- app.bundle.js

5.2.2. Средства запуска

Для запуска программы использовался веб-браузер Firefox Developer Edition версии 101.0b5

5.3. Порядок проведения испытаний

Программа запускается на устройстве, отвечающем требованиям к составу и параметрам технических средств.

Последовательно проверяются:

- Функциональные требования
- Требованиям к информационной и программной совместимости
- Требованиям к надежности

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

6. Методы испытаний

6.1. Функциональные требования

Последовательно проверяются функциональные требования.

Алгоритм Фараха

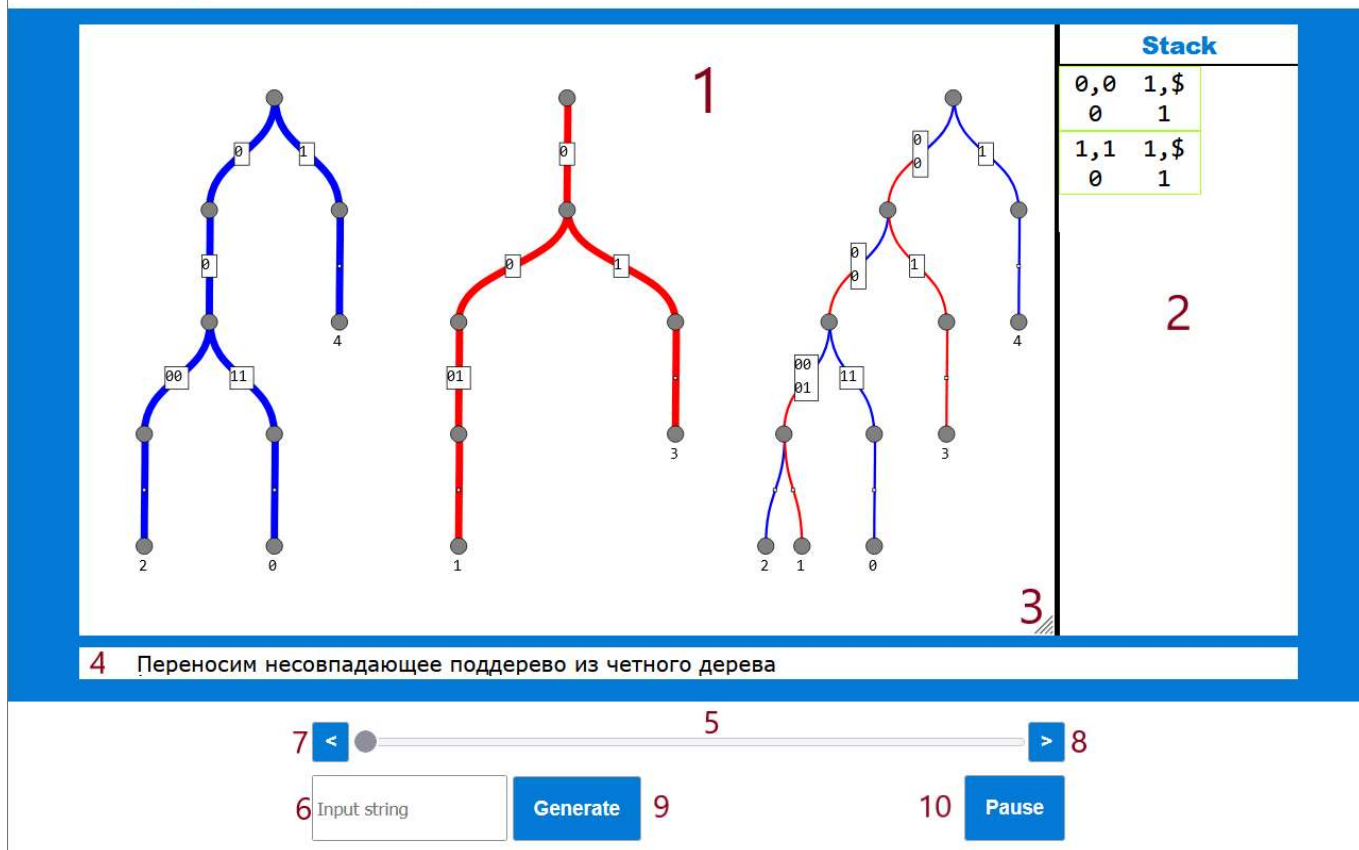


Рисунок 1. Интерфейс приложения

1. Поле ввода для строки, по которой будет строиться сжатое суффиксное дерево.

Выполнено. Находится на рис. 1 под номером (6).

2. Поэтапная визуализация алгоритма Фараха

2.1. Разбиение строки на пары символов и дальнейшее сжатие.

2.1.1. Манипуляции с символами и парами символов показываются при помощи перемещения текстовых элементов.

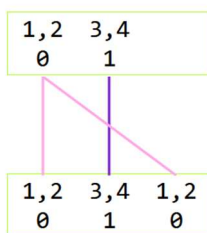


Рисунок 2. Массив пар символов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

2.2. Построение четного суффиксного дерева.

2.2.1. Дерево состоит из вершин и ребер. Вершина показывается в виде круга. Ребро показывается в виде отрезка, соединяющего вершины. Ребро подписывается текстом, который оно обозначает.

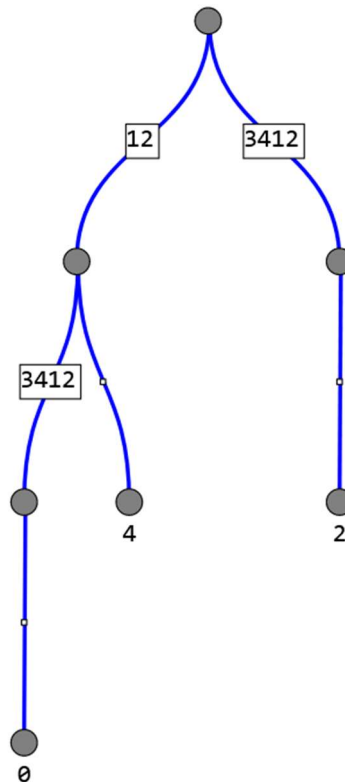


Рисунок 3. Четное дерево

2.3. Построение нечетного суффиксного дерева.

2.3.1. Требования к дереву аналогичны требованиям в пункте “Построение четного суффиксного дерева”.

2.3.2. Ребра четного дерева имеют цвет, отличный от цвета ребер нечетного дерева.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

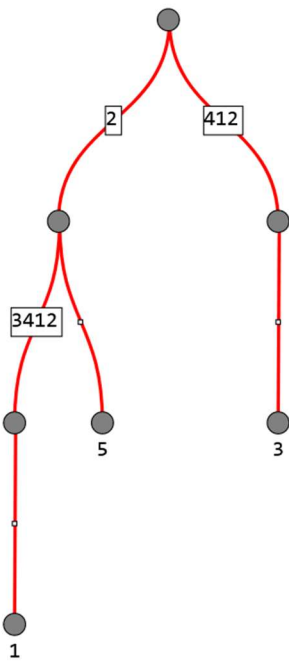


Рисунок 4. Нечетное дерево

2.4. Слияние деревьев.

2.4.1. Требования к дереву аналогичны требованиям в пункте “Построение четного суффиксного дерева”.

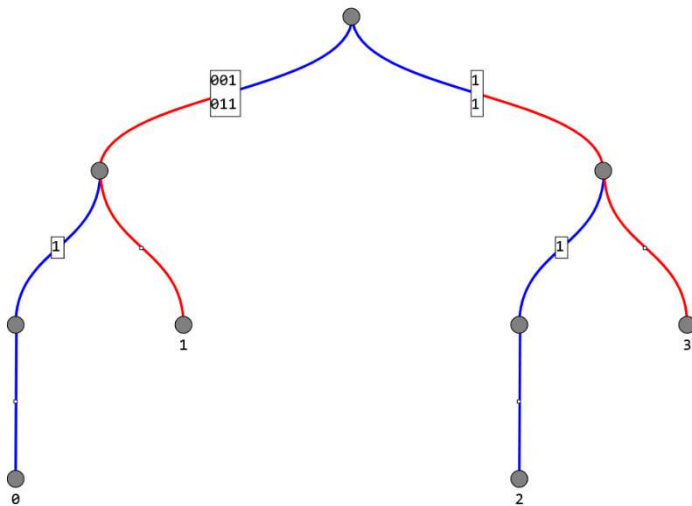


Рисунок 5. Результат слияния деревьев

2.5. Удаление двойных ребер.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

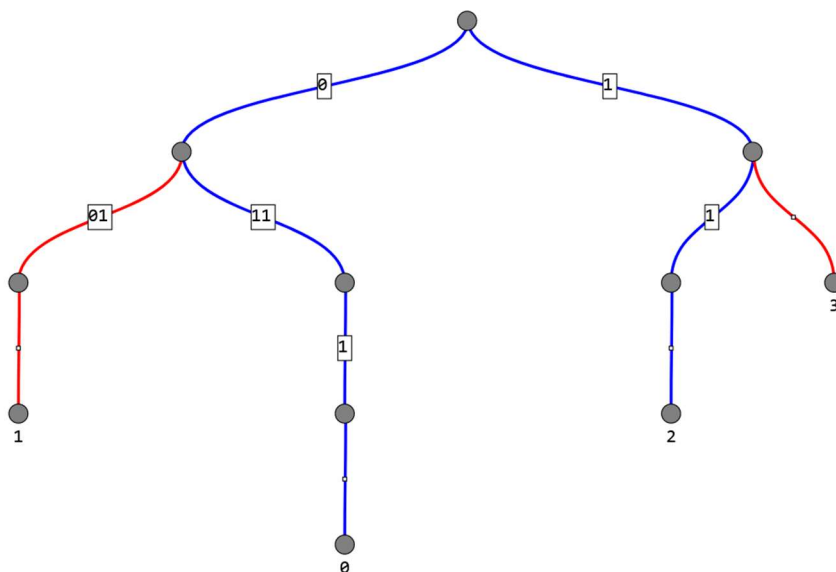


Рисунок 6. Дерево после удаления двойных ребер

3. Возможность приостановить визуализацию.

Выполнено.

Есть возможность запуска автоматического воспроизведения визуализации при помощи кнопки (10). Также эта кнопка (10) выполняет функцию приостановки автоматического воспроизведения визуализации. Ее назначение меняется в зависимости от того, воспроизводится ли визуализация автоматически в данный момент.

4. Возможность перейти к следующему кадру визуализации.

Выполнено.

5. Возможность перейти к предыдущему кадру визуализации.

Выполнено.

Перемещение между кадрами осуществляется при помощи кнопок “Шаг назад” (7) и “Шаг вперед” (8). Также для этого можно использовать кнопки “Стрелка влево” и “Стрелка вправо” на клавиатуре.

6. Возможность построения сжатого суффиксного дерева для новой строки без перезапуска приложения.

Выполнено.

Существует возможность перезапуска визуализации для другой строки без перезапуска программы. Для этого нужно ввести новую строку в поле ввода (6), после чего нажать кнопку “Сгенерировать визуализацию” (9)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

Функциональные требования выполнены.

6.2. Требования к информационной и программной совместимости

Последовательно проверяются требования к информационной и программной совместимости.

6.2.1. Требования к информационным структурам на входе.

Есть возможность ввести входные данные в виде строки в поле ввода (6).

6.2.2. Требования к информационным структурам на выходе.

Каждый этап алгоритма визуализируется. Переходы между шагами, при возможности, сопровождаются анимацией.

Данные и структуры данных, использованные во время работы алгоритма, представляются пользователю в удобном для восприятия виде.

Последний шаг визуализации отображает построенное сжатое суффиксное дерево для введенной пользователем строки.

6.2.3. Требования к исходным кодам и языкам программирования.

Исходный код программы написан на языках программирования HTML, CSS и TypeScript. Настроена сборка и транспилиция исходного кода на TypeScript в JavaScript с целью дальнейшего запуска программы в браузере.

6.3. Требования к надежности.

6.3.1. При попытке запустить алгоритм на пустой строке, будет выведено сообщение о некорректных входных данных. Программа не завершается аварийно.

6.3.2. После загрузки на устройство программа имеет возможность корректной работы без соединения с сетью интернет.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. № дубл.	Подп. и дата

Лист регистрации изменений

[illegible]