

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №1

Вариант №5

Выполнил:

Конопля А.К.

Проверил:

Мусаев А.А.

Санкт-Петербург

2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ПОСТАНОВКА ЗАДАЧИ.....	4
ХОД РАБОТЫ.....	5
Задание 1. Алгоритмы поиска подстроки.....	5
Наивный алгоритм	5
Алгоритм Рабина-Карпа.....	6
Алгоритм Бойера-Мура.....	6
Алгоритм Кнута-Мориса-Пратта	7
Задание 2. Антиплагиат.....	8
ЗАКЛЮЧЕНИЕ.....	9
ПРИЛОЖЕНИЕ.....	10

ВВЕДЕНИЕ

Данная лабораторная работа посвящена изучению алгоритмов поиска подстроки.

ПОСТАНОВКА ЗАДАЧИ

1) Заполните массив 500 числами (четный вариант – простые числа, нечетный вариант – числа Фибоначчи) написанными слитно. Используя каждый изученный алгоритм поиска подстроки (наивный, Рабина-Карпа, Бойера-Мура, Кнута-Морриса-Пратта), посчитайте количество наиболее часто встречающихся двузначных чисел в образовавшейся строке. Сравните изученные алгоритмы поиска подстроки. Сделайте вывод о их достоинствах и недостатках.

2) Дан набор рефератов. Выберите любой алгоритм поиска и определите количество плагиата (в % от общего количества символов в реферате) в тексте реферата, взяв за основу соответствующие статьи из Википедии (название файла = название статьи). За плагиат считать любые 3 совпавших слова, идущих подряд. Обоснуйте выбранный алгоритм поиска.

ХОД РАБОТЫ

Задание 1. Алгоритмы поиска подстрок

В данном задании необходимо реализовать алгоритмы поиска подстрок (Наивный, Рабина-Карпа, Бойера-Мура, Кнутта-Морриса-Пратта), посчитать количество наиболее часто встречающихся в строке, состоящей из 500 чисел Фибоначчи. Проанализировать достоинства и недостатки.

Наивный алгоритм

Основывается на переборе всех допустимых сдвигов строки.

Достоинства:

- 1) Требует $O(1)$ памяти.
- 2) Простая и понятная реализация.

Недостатки:

- 1) Требует $O(m * (n - m))$ операций, вследствие чего алгоритм работает медленно, в случае если длина паттерна велика.

Результат выполнения программы:

Число 71 встречается чаще всего, 296 раз

Рисунок 1. Результат выполнения наивного алгоритма

Алгоритм Рабина-Карпа

Основывается на поиске шаблона, использующий хэширование сравниваемой строки.

Достоинства:

- 1) Высокая скорость работы – $O(n + m)$, где n – длина строки, а m – длина образца.

Недостатки:

- 1) Скорость функции зависит от функции хэширования строки

Результат выполнения работы:

Число 71 встречается чаще всего, 296 раз

Рисунок 2. Результат выполнения алгоритма Рабина-Карпа

Алгоритм Бойера-Мура

Алгоритм Бойера-Мура основана на том, что ценой некоторого количества предварительных вычислений над шаблоном (но не над строкой, в которой ведётся поиск), шаблон сравнивается с исходным текстом не во всех позициях — часть проверок пропускается как заведомо не дающая результата.

Достоинства:

- 1) Алгоритм Бойера-Мура на хороших данных очень быстр, а вероятность появления плохих данных крайне мала. Поэтому он оптимален в большинстве случаев, когда нет возможности провести предварительную обработку текста, в котором проводится поиск.

Недостатки:

- 1) На больших алфавитах (например, Юникод) может занимать много памяти. В таких случаях либо обходятся хэш-таблицами, либо дробят алфавит, рассматривая, например, 4-байтовый символ как пару двухбайтовых.

Результат выполнения программы

Число 71 встречается чаще всего, 296 раз

Рисунок 3. Результат выполнения алгоритма Бойера-Мура

Алгоритм Кнута-Мориса-Пратта

Алгоритм Кнута-Мориса-Пратта основан на том, что за счёт некоторого количества предварительных вычислений можно рассчитать наиболее выгодную длину пропуска при несовпадении, так называемой префикс функции.

Достоинства:

- 1) Временная сложность алгоритма одна из наименьших в сравнении с другими алгоритмами – $O(P + T)$ P – сложность префикс функции, T – сложность самой функции.

Недостатки:

- 2) Алгоритм сложен в понимании

Результат выполнения программы:

Число 71 встречается чаще всего, 296 раз


Рисунок 4. Результат выполнения алгоритма Кнута-Мориса-Пратта

Вывод: Были реализованы различные алгоритмы поиска подстрок. Наиболее быстрый алгоритм поиска подстрок – алгоритм Кнута-Мориса-Пратта, этот алгоритм кратно опережает по времени любой другой.

Задание 2. Антиплагиат

В данном задании необходимо написать алгоритм, высчитывающий оригинальность текста, сравнивая написанный текст с текстом статьи на сайте Wikipedia. Данное задание реализовано за счёт алгоритма Кнута-Мориса-Пратта. Были протестированы все алгоритмы для выполнения данной задачи и выбранный алгоритм показал себя лучше всего при наличии образца подстроки с повторениями, а в словах русского языка есть множество повторяющихся символов, так же мог быть использован алгоритм Бойера-Мура, но он показывает себя хуже в случаях, описанных выше.

Результат выполнения программы:



34.43 %

Рисунок 5. Результат выполнения программы «антиплагиат»

Вывод: Была реализована программа антиплагиат, на основе алгоритма поиска подстрок Кнута-Мориса-Пратта.

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы были изучены и реализованы алгоритмы поиска подстрок, а также реализована программа антиплагиат, основывающаяся на алгоритме Кнута-Мориса-Пратта.

ПРИЛОЖЕНИЕ

Приложение А.

<https://github.com/AlexeyKonoplia/lab1>