

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №10**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема. Поиск в тесте образца. Алгоритмы. Эффективность алгоритмов.

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы ИКБО-06-22 |  | Кликушин В.И. |
| Принял старший преподаватель |  | Скворцова Л.А. |

Москва 2023

ОГЛАВЛЕНИЕ

[1 ОТЧЁТ ПО ЗАДАНИЮ 1 3](#_Toc136980668)

[1.1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc136980669)

[1.2 ОПИСАНИЕ МОДЕЛИ (ПОДХОД К РЕШЕНИЮ) 3](#_Toc136980670)

[1.3 КОД ФУНКЦИИ 3](#_Toc136980671)

[1.4 ТАБЛИЦЫ ТЕСТИРОВАНИЯ 5](#_Toc136980672)

[1.5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ 6](#_Toc136980673)

[1.6 ТЕОРЕТИЧЕСКАЯ СЛОЖНОСТЬ АЛГОРИТМОВ 7](#_Toc136980674)

[1.7 ТАБЛИЦЫ РЕЗУЛЬТАТОВ 8](#_Toc136980675)

[1.8 ВЫВОДЫ 8](#_Toc136980676)

[2 ОТЧЁТ ПО ЗАДАНИЮ 2 9](#_Toc136980677)

[2.1 ПОСТАНОВКА ЗАДАЧИ 9](#_Toc136980678)

[2.2 РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ 9](#_Toc136980679)

[2.3 КОД ФУНКЦИИ 9](#_Toc136980680)

[2.4 ТАБЛИЦА ТЕСТИРОВАНИЯ 9](#_Toc136980681)

[2.5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ 10](#_Toc136980682)

[3 ВЫВОДЫ 12](#_Toc136980683)

[4 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ 13](#_Toc136980684)

# 1 ОТЧЁТ ПО ЗАДАНИЮ 1

## 1.1 ПОСТАНОВКА ЗАДАЧИ

Разработать и реализовать алгоритм Рабина-Карпа для определения количества вхождения каждого слова в текст.

Задание варианта: Дан текст и список слов. Определить, сколько раз каждое слово входит в текст.

## 1.2 ОПИСАНИЕ МОДЕЛИ (ПОДХОД К РЕШЕНИЮ)

Разработать вспомогательную функцию, которая будет разбивать введенный список слов (представлен строкой, содержащей слова, записанные через пробел) на отдельные слова и для каждого слова будет вызывать функцию поиска Рабина – Карпа, которая в свою очередь будет сообщать о количестве вхождений данного слова в исходный текст.

Пример: i secret car – строка, содержащая три слова, записанные через пробел, которые необходимо найти в исходном тексте.

Алгоритм Рабина – Карпа использует хеширование для поиска подстроки. Вычисляется хеш подстроки, затем хеш первого фрагмента текста. Далее проходимся по тексту и ищем совпадения, при совпадении хеш – функций необходимо посимвольно сравнить две подстроки.

## 1.3 КОД ФУНКЦИИ

Предусловие: sub – подстрока, text – исходный текст.

Постусловие: количество вхождений подстроки sub в текст text.

int Rabin\_Karp(string sub, string text)

{

int q = 127, p = 0, t = 0, h = 1;

int cnt = 0;

// Вычисляем хеш-функцию sub и первую хеш-функцию text

for (int i = 0; i < sub.size() - 1; i++)

h = (h \* d) % q;

for (int i = 0; i < sub.size(); i++)

{

p = (d \* p + sub[i]) % q;

t = (d \* t + text[i]) % q;

}

// Проходимся по тексту и ищем совпадения хеш-функций

for (int i = 0; i <= text.size() - sub.size(); i++)

{

// Если хеш-функции совпадают, проверяем посимвольно

if (p == t)

{

int j;

for (j = 0; j < sub.size(); j++)

{

if (text[i + j] != sub[j])

break;

}

if (j == sub.size()) //Если все символы совпали, увеличиваем счетчик вхождений слова в текст

cnt++;

}

// Вычисляем хеш-функцию следующей подстроки текста

if (i < text.size() - sub.size())

{

t = (d \* (t - text[i] \* h) + text[i + sub.size()]) % q;

// Если результат вычисления хеш-функции отрицательный, добавляем q

if (t < 0)

t = (t + q);

}

}

return cnt;

}

Предусловие: str – исходный текст.

Постусловие: выводит сообщения о количестве вхождений каждого слова из списка в исходный текст.

void search(string str)

{

cout << " Введите слова через пробел: \n";

string words;

cin.ignore();

getline(cin, words);

string temp;

for (int i = 0; i < words.length(); i++)

{

if (words[i] != ' ')

{

temp += words[i];

}

else

{

cout << " Слово " << temp << " встретилось в исходном тексте " << Rabin\_Karp(temp, str) << " раз" << endl;

temp = "";

}

}

if (temp.length() != 0)

{

cout << " Слово " << temp << " встретилось в исходном тексте " << Rabin\_Karp(temp,str) << " раз" << endl;

}

}

## 1.4 ТАБЛИЦЫ ТЕСТИРОВАНИЯ

|  |  |  |
| --- | --- | --- |
| Номер теста | Входные данные | Эталон результата |
| 1 | sub = "i"  text = "my name is vlad i am 18 yo i like codding" | Слово i встретилось в исходном тексте 5 раз |
| 2 | sub = "secret Secret"  t = "my secret is my secret don’t tell the secret" | Слово Secret встретилось в исходном тексте 0 раз  Слово secret встретилось в исходном тексте 3 раз |
| 3 | sub = "string"  text = "include string and string.h" | Слово string встретилось в исходном тексте 2 раз |
| 4 | sub = "абсолютно абсолютный"  text = "абсолютная чемпионка по теннису абсолют…" | Слово абсолютно встретилось в исходном тексте 0 раз  Слово абсолютный встретилось в исходном тексте 0 раз |

## 1.5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 – Результат теста 1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 – Результат теста 2

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Результат теста 3

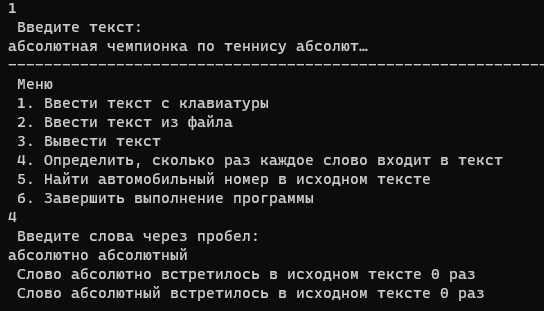


Рисунок 4 -Результат теста 4

## 1.6 ТЕОРЕТИЧЕСКАЯ СЛОЖНОСТЬ АЛГОРИТМОВ

n – длина текста, m – длина подстроки, x – позиция первого вхождения

Таблица 1. Оценка сложности алгоритма Рабина Карпа

|  |  |  |
| --- | --- | --- |
| Оператор | Количество сравнений при неудачном поиске | Количество сравнений для нахождения первого вхождения |
| q ← 127 |  |  |
| p ← 0 |  |  |
| t ← 0 |  |  |
| h ← 1 |  |  |
| cnt ← 0 |  |  |
| for i ← 0 to sub.size()-1 do | m | m |
| h ← (h \* d) % q |  |  |
| od |  |  |
| for i ← 0 to sub.size() do | m+1 | m+1 |
| p ← (d \* p + sub[i]) % q |  |  |
| t ← (d \* t + text[i]) % q |  |  |
| od |  |  |
| for i ← 0 to text.size() -sub.size() -1 do | n-m | x |
| if p == t then | n-m-1 | x |
| j ← 0 |  |  |
| for j ← 0 to sub.size() do |  | m |
| if (text[i+j] != sub[j]) then |  | m |
| break |  |  |
| if j==sub.size() then |  | 1 |
| cnt ← cnt + 1 |  |  |
| if i<text.size()-sub.size() then | n-m-1 | x |
| t ← (d \* (t - text[i] \* h) + text[i + sub.size()]) % q |  |  |
| if t < 0 then | n-m-1 |  |
| t ← t + q |  |  |
| od |  |  |
| Возврат cnt |  |  |

Итого: сравнений при неуспешном поиске = 4n-2m-2;

4m +3x + 2 сравнений для нахождения первого вхождения подстроки

## 1.7 ТАБЛИЦЫ РЕЗУЛЬТАТОВ

Таблица 2. Анализ алгоритма Рабина Карпа

|  |  |  |  |
| --- | --- | --- | --- |
| n | T(n) - время | Tэт = f(C) = 4n-2m-2 | Tэп = Cф |
| 100 | 0.0041 мс | 388 | 395 |
| 1000 | 0.0337 мс | 3988 | 3995 |
| 10000 | 0.3704 мс | 39988 | 39995 |
| 100000 | 3.6987 мс | 399988 | 399995 |
| 1000000 | 36.9 мс | 3999988 | 3999995 |

## 1.8 ВЫВОДЫ

Алгоритм редко используется для поиска одиночного образца, но имеет значительную теоретическую важность и очень эффективен в поиске совпадений множественных образцов одинаковой длины. Одно из простейших практических применений алгоритма Рабина — Карпа состоит в определении плагиата. Алгоритм Рабина — Карпа пытается ускорить проверку эквивалентности образца с подстроками в тексте, используя хеш-функцию.

# 2 ОТЧЁТ ПО ЗАДАНИЮ 2

## 2.1 ПОСТАНОВКА ЗАДАЧИ

Дан текст, содержащий стандартные российские автомобильные номера, формируемые по формату: «буква - три цифры - две буквы - код региона». Причем код региона может быть двух или трехзначным, а в качестве букв применяются только те, что похожи внешне на латиницу. Номер в тексте ограничен с двух сторон пробелом. Найти автомобильный номер в исходном тексте.

## 2.2 РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ

Регулярное выражение: ("[АВСЕКНМОРТХУ]\\d{3}[АВСЕКНМОРТХУ]{2}\\d{2,3}"

## 2.3 КОД ФУНКЦИИ

void car\_number(string str)

{

regex r("[АВСЕКНМОРТХУ]\\d{3}[АВСЕКНМОРТХУ]{2}\\d{2,3}");

smatch m;

regex\_search(str, m, r);

for (auto x : m)

{

cout << x << endl;

}

}

## 2.4 ТАБЛИЦА ТЕСТИРОВАНИЯ

|  |  |  |
| --- | --- | --- |
| Номер теста | Входные данные | Эталон результата |
| 1 | str = "The car’s number was А111АА11" | А111АА11 |
| 2 | str = "Which number is correct У345УУУ22 or У345УВ164" | У345УВ164 |
| 3 | str = "БМВ Х3 В123АУ777 черн.Е83 2.0d" | В123АУ777 |

## 2.5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – Результат теста 1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – Результат теста 2

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – Результат теста 3

# 3 ВЫВОДЫ

В ходе выполнения практической работы мной были получены навыки в разработке и использовании алгоритма Рабина Карпа поиска подстроки в тексте; а также навыки построение и использования регулярных выражений для поиска шаблонных подстрок в тексте.

# 4 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Учебно-методическое пособие СиАОД (часть 1)

2. Приложение к практическим работам – СДО (online-edu.mirea.ru)