

Государственное образовательное учреждение высшего профессионального образования
“Московский государственный технический университет имени Н.Э.Баумана”

Дисциплина: АНАЛИЗ АЛГОРИТМОВ

ЛАБОРАТОРНАЯ РАБОТА № 7

Алгоритмы сравнения с образцом

Студент:

Сиденко Анастасия Геннадьевна

Группа: ИУ7-53Б

Преподаватели:

Строганов Юрий Владимирович

Волкова Лилия Леонидовна

2019 г.

Содержание

Введение	2
1 Аналитическая часть	3
1.1 Описание задачи	3
1.2 Пути решения	3
1.3 Выводы	4
2 Конструкторская часть	5
2.1 Функциональная модель	5
2.2 Схемы алгоритмов	5
2.3 Выводы	7
3 Технологическая часть	8
3.1 Требования к программному обеспечению	8
3.2 Средства реализации	8
3.3 Листинг кода	8
3.4 Тестирование	8
3.5 Выводы	9
4 Экспериментальная часть	10
4.1 Примеры работы	10
4.2 Результаты тестирования	10
4.3 Выводы	10
Заключение	11

Введение

Те, кому приходится часто работать с текстовыми редакторами, часто пользуются функцией нахождения нужных слов в тексте, которая существенно облегчает редактирование документов и поиск нужной информации.

Действительно, современные программы обработки текста включают в свой функционал поиск и замену текстовых фрагментов[1].

Конечно, сейчас функции поиска входят во многие языки программирования высокого уровня – чтобы найти строчку в небольшом тексте используется встроенная функция[2].

Цель данной работы изучить и разработать алгоритм поиска подстроки в строке.

Задачи данной работы.

1. Изучить основные алгоритмы, решающих задачу поиска.
2. Реализовать данные алгоритмы.

1 Аналитическая часть

Необходимо изучить алгоритмы поиска подстроки в строке.

1.1 Описание задачи

Поиск строки формально определяется следующим образом. Пусть задан массив S элементов и массив X элементов. Поиск строки обнаруживает первое вхождение X в S , результатом будем считать индекс i , указывающий на первое с начала строки (с начала массива S) совпадение со словом.

1.2 Пути решения

1. Алгоритм Кнута-Морриса-Пракса

Алгоритм был разработан Кнутом и Праксом и независимо от них Моррисом в 1977 г.

После частичного совпадения начальной части подстроки X с соответствующими символами строки S мы фактически знаем пройденную часть строки и можем «вычислить» некоторые сведения (на основе самого подстроки X), с помощью которых потом быстро продвинемся по тексту.

Идея КМП-поиска – при каждом несовпадении двух символов текста и образа образ сдвигается на самое длинное совпадение начала с концом префикса (не учитывая тривиальное совпадение самого с собой) [3].

Пример.

Создается массив сдвигов, таблица 1.

0	1	2	3	4	5
a	b	c	a	b	d
0	0	0	1	2	0

Таблица 1. Массив сдвигов.

В таблице 2 представлена работа алгоритма.

строка	a	b	c	a	b	e	a	b	c	a	b	c	a	b	d
подстрока	a	b	c	a	b	d									
подстрока				a	b	c	a	b	d						
подстрока						a	b	c	a	b	d				
подстрока							a	b	c	a	b	d			
подстрока										a	b	c	a	b	d

Таблица 2. Алгоритм КМП.

2. Алгоритм Бойера-Мура

Алгоритм поиска строки Бойера – Мура считается наиболее быстрым среди алгоритмов общего назначения, предназначенных для поиска подстроки в строке. Был разработан Бойером и Муром в 1977 году. Преимущество этого алгоритма в 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 пропускаются как заведомо не дающие результата.

Основная идея алгоритма – начать поиск не с начала, а с конца подстроки. Наткнувшись на несовпадение, мы просто смещаем подстроку до самого правого вхождения данного символа, не учитывая последний [3].

Пример.

Создается массив прыжков, таблица 3.

a	b	c	a	b	d
2	1	3	2	1	6

Таблица 3. Массив прыжков.

В таблице 4 представлена работа алгоритма.

строка	a	b	c	a	b	e	a	b	c	a	b	c	a	b	d
подстрока	a	b	c	a	b	d									
подстрока							a	b	c	a	b	d			
подстрока										a	b	c	a	b	d

Таблица 4. Алгоритм БМ.

1.3 Выводы

В данной работе стоит задача реализации алгоритмов поиска подстроки в строке.

2 Конструкторская часть

2.1 Функциональная модель

На рисунке 1 представлена функциональная модель нашей задачи.



Рис. 1 - Функциональная модель алгоритма нахождения вхождения подстроки в строку

2.2 Схемы алгоритмов

Приведем схемы алгоритмов (см. рисунки 2 и 3).

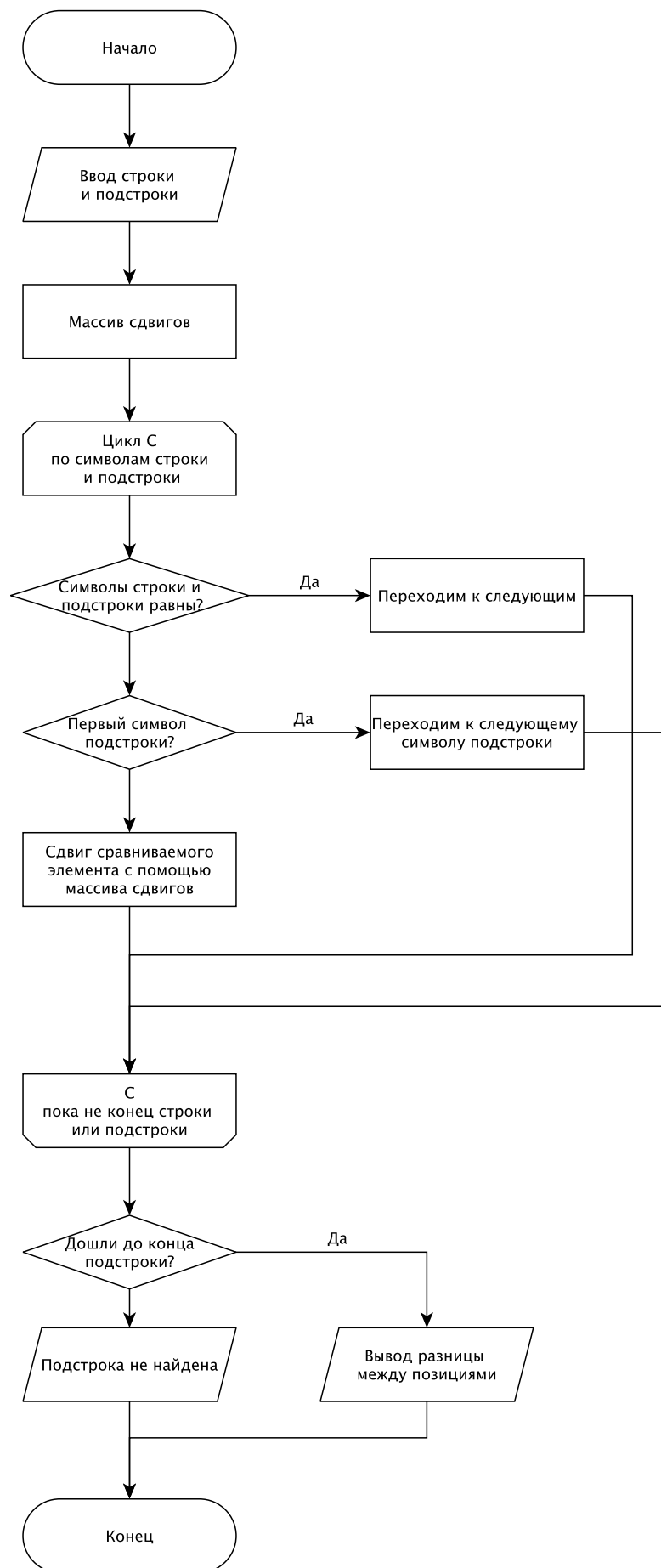


Рис. 2 - Алгоритм КМП

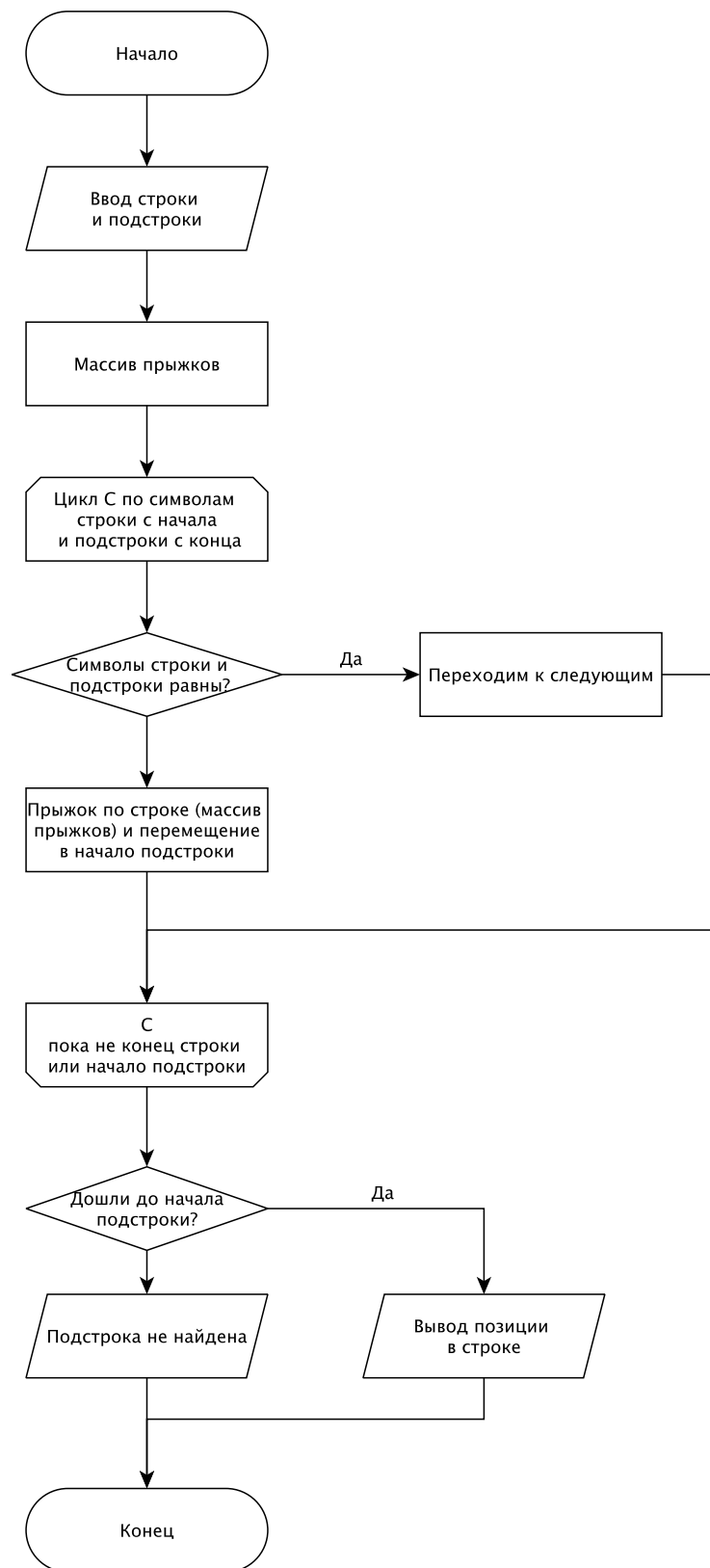


Рис. 3 - Алгоритм БМ

2.3 Выводы

Алгоритмы изучены, необходимо приступить к реализации и тестированию.

3 Технологическая часть

Стоит задача реализации алгоритмов сравнения с образцом на основании автоматов, а именно Кнута - Морриса - Пратта и Бойера - Мура.

3.1 Требования к программному обеспечению

ПО должно предоставлять возможность разработки и запуска программы.

3.2 Средства реализации

В качестве языка программирования был выбран C++, так как я знакома с этим языком программирования и он удовлетворяет требованиям.

3.3 Листинг кода

На листингах 1 и 2 представлен код алгоритмов сравнения с образцом.

Листинг 1: Алгоритм Кнута - Морриса - Пратта (КМП)

```
1 d = shift_array(x)
2 i = j = 0
3 while i < len(s) and j < len(x):
4     if x[j] == s[i]:
5         i += 1
6         j += 1
7     elif 0 == j:
8         i += 1
9     else:
10        j = d[j - 1]
11
12 if j == len(x):
13     return i - j + 1
14 return -1
```

Листинг 2: Алгоритм Бойера - Мура

```
1 d = jump_array(x)
2 i = j = k = len(x)
3 while k <= len(s) and j > 0:
4     if x[j - 1] == s[i - 1]:
5         j -= 1
6         i -= 1
7     else:
8         k += d[ord(s[i - 1])]
9         i = k
10        j = len(x)
11
12 if j <= 0:
13     return i + 1
14 else:
15     return -1
```

3.4 Тестирование

В таблице 5 представлена заготовка данных для тестирования заданных алгоритмов.

Строка1	Строка2	Ожидаемый результат
aaaaa	a	1
ааро	ро	3
роаа	ро	1
qwerty	a	Подстрока не найдена
ророророро	ро	1

Таблица 5. Подготовленные тестовые данные.

3.5 Выводы

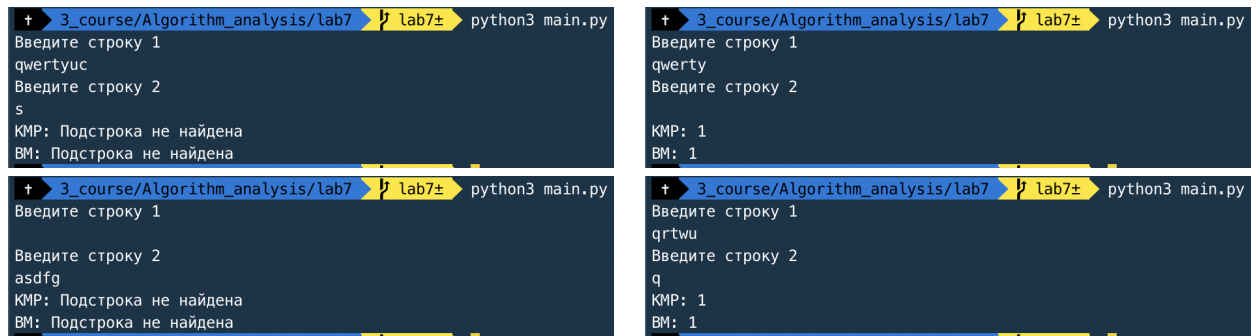
Алгоритмы были изучены и реализованы, необходимо приступить к тестированию.

4 Экспериментальная часть

Необходимо протестировать полученную программу.

4.1 Примеры работы

На рисунке 4 представлены примеры работы программы на разных входных данных.



The figure consists of four terminal screenshots arranged in a 2x2 grid. Each screenshot shows the execution of a Python script named 'main.py' in a terminal window. The prompt is 'python3 main.py'. The program prompts the user to enter two strings, 'Введите строку 1' and 'Введите строку 2'. The results for each input pair are shown below the prompts.

Введите строку 1	Введите строку 2	КМР:	БМ:
qwertyuc	s	Подстрока не найдена	Подстрока не найдена
qwerty	qwerty	1	1
qgtwu	q	1	1

Рис. 4 - Примеры работы

4.2 Результаты тестирования

Проверяем нашу программу на тестах из таблицы 5. Полученные результаты представлены в таблице 6.

Строка1	Строка2	КМП	БМ
aaaaa	a	1	1
aapo	po	3	3
poaa	po	1	1
qwerty	a	Подстрока не найдена	Подстрока не найдена
pororororo	po	1	1

Таблица 6. Тестирование программы.

Тесты пройдены

4.3 Выводы

Изучены, реализованы и протестированы алгоритмы сравнения с образцом на основании автоматов.

Заключение

В данной работе изучены, разработаны и протестированы алгоритм поиска подстроки в строке, такие как Кнута-Морриса-Пратта и Бойера-Мура.

Решены следующие задачи.

1. Изучены основные алгоритмы, решающих задачу поиска.
2. Реализованы необходимые алгоритмы.

Список литературы

- [1] [Электронный ресурс]. - Режим доступа: <https://support.office.com/ru-ru/article/Поиск-текста-в-документе-672d56af-7ad9-4b98-872c-ceed9c81c21c> (дата обращения: 13.12.2019)
- [2] [Электронный ресурс]. - Режим доступа: <https://pythonworld.ru/tipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html> (дата обращения: 13.12.2019)
- [3] Семинары