



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе №7
по курсу «Анализ Алгоритмов»
на тему: «Алгоритмы поиска»

Студент ИУ7-51Б
(Группа)

(Подпись, дата)

Савинова М. Г.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Строганов Ю. В.
(Фамилия И. О.)

2024 г.

СОДЕРЖАНИЕ

1	Аналитический раздел	4
1.1	Алгоритм Кнута – Морриса – Пратта	4
1.2	Модификация алгоритма Кнута – Морриса – Пратта	4
2	Конструкторский раздел	6
2.1	Требования к программному обеспечению	6
2.2	Разработка алгоритмов	6
3	Технологический раздел	10
3.1	Средства реализации	10
3.2	Сведения о модулях программы	10
3.3	Реализация алгоритмов	10
3.4	Тестирование	10
4	Исследовательский раздел	11
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

Введение

Задача поиска в строке заключается в следующем: дана длинная строка («текст») $w = a_1 \dots a_n$, и короткая искомая строка («шаблон») $x = b_1 \dots b_m$. Требуется найти все вхождения x в w в качестве подстроки, то есть, все смещения s , для которых подстрока $ws = a_{s+1} \dots a_{s+m}$ совпадает с $b_1 \dots b_m$ [1].

Целью данной лабораторной работы является исследование лучших и худших случаев работы алгоритмов поиска.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) описать используемые алгоритмы поиска;
- 2) выбрать средства программной реализации;
- 3) реализовать данные алгоритмы поиска;
- 4) проанализировать алгоритмы по количеству сравнений.

1 Аналитический раздел

В данном разделе будут рассмотрены алгоритм Кнута – Морриса – Пратта и его модификация.

Для этого введем следующие обозначения:

- подстрока x называется *префиксом* строки w , если есть такая подстрока z , что $w = xz$;
- подстрока x называется *суффиксом* строки w , если есть такая подстрока z , что $w = zx$ [2].

1.1 Алгоритм Кнута – Морриса – Пратта

Рассматриваемый алгоритм основывается на том, что после частичного совпадения начальной части подстроки с соответствующими символами строки фактически известна пройденная часть строки и можно, вычислить некоторые сведения, с помощью которых затем быстро продвинуться по строке.

Основным отличием алгоритма Кнута – Морриса – Пратта от алгоритма прямого поиска заключается в том, что сдвиг подстроки выполняется не на один символ на каждом шаге алгоритма, а на некоторое переменное количество символов. Следовательно, перед тем как осуществлять очередной сдвиг, необходимо определить величину сдвига [2].

Для определения этого сдвига используют *префиксную функцию*. Это функция, которая для всякого префикса $b_1 \dots b_i$ строки $x = b_1 \dots b_m$ выдает длину наибольшего суффикса подстроки $b_1 \dots b_i$, который одновременно будет и префиксом x [1].

1.2 Модификация алгоритма Кнута – Морриса – Пратта

Модификация заключается в введении эвристики «плохого» символа по аналогии с алгоритмом Бойера – Мура с адаптацией к обходу подстроки слева направо.

Эвристика «плохого» символа определяет сдвиг наибольшего возможного количества позиций, если символ строки не совпадает с символом подстроки в заданной позиции [1].

Таким образом, в модифицированном алгоритме мы используем как информацию о суффиксах, так и о длине сдвига.

Вывод

В данном разделе были рассмотрены алгоритм Кнута – Морриса – Пратта и его модификация.

2 Конструкторский раздел

В данном разделе будут представлены псевдокоды алгоритмов Кнута – Морриса – Пратта и его модификации.

2.1 Требования к программному обеспечению

К программному обеспечению предъявлен ряд требований:

- 1) наличие интерфейса для выбора действия;
- 2) возможность ввода строки и подстроки;
- 3) возможность выполнения операции поиска подстроки в строке.

2.2 Разработка алгоритмов

В качестве инструмента для создания псевдокода использован пакет `algorithm`.

В случае нахождения подстроки в строке просиходит возврат **shift**, то есть смещение относительно начала строки, иначе — **-1**.

Определим следующие операторы и функции:

- оператор \leftarrow обозначает присваивание значение переменной;
- оператор $[i]$ обозначает получение элемента из массива с индексом i ;
- функция `max` возвращает максимальное значение;
- функция `ord` возвращает целое число, представляющее символ Юникода;
- функция `length` возвращает длину массива, строки;

В листингах 1–2 рассмотрены псевдокоды алгоритмов формирования дополнительных массивов суффиксов и сдвигов соответственно.

В листингах 3–4 рассмотрены псевдокоды алгоритма Кнута – Морриса – Пратта и его модификации.

Листинг 1 Псевдокод алгоритма формирования массива суффиксов

На входе: строка $w = a_1 \dots a_n$, ссылка на массив π

```
1: function MAKEPiLIST( $w$ )
2:    $j \leftarrow 0$ 
3:    $i \leftarrow 1$ 
4:   while  $i < n$  do
5:     if  $a_j = a_i$  then
6:        $\pi[i] \leftarrow j + 1$ 
7:        $j \leftarrow j + 1$ 
8:        $i \leftarrow i + 1$ 
9:     else
10:      if  $j = 0$  then
11:         $\pi[i] \leftarrow 0$ 
12:         $i \leftarrow i + 1$ 
13:      else
14:         $j \leftarrow \pi[j - 1]$ 
15:      end if
16:    end if
17:  end while
18: end function
```

Листинг 2 Псевдокод алгоритма формирования массива сдвигов

На входе: строка $w = a_1 \dots a_n$, ссылка на массив *badChars*

```
1: function MAKEBADCHARS( $w$ , badChars)
2:    $j \leftarrow 0$ 
3:    $i \leftarrow 1$ 
4:   for  $i = (0, n)$  do
5:      $\text{badChars}[\text{ord}(w_i)] \leftarrow i$ 
6:   end for
7: end function
```

Листинг 3 Псевдокод алгоритма Кнута – Морриса – Пратта

На входе: строка $w = a_1 \dots a_n$, подстрока $x = b_1 \dots b_m$, массив суффиксов π

```
1: function KMP( $w, x, \pi$ )
2:    $i \leftarrow 0$ 
3:    $j \leftarrow 0$ 
4:    $n \leftarrow \text{length}(w)$ 
5:    $m \leftarrow \text{length}(x)$ 
6:   while  $i < n$  do
7:     if  $a_i = b_j$  then
8:        $i \leftarrow i + 1$ 
9:        $j \leftarrow j + 1$ 
10:    if  $j = m$  then return  $i - n$ 
11:    end if
12:    else
13:      if  $j = 0$  then
14:         $i \leftarrow i + 1$ 
15:      else
16:         $j \leftarrow \pi[j - 1]$ 
17:      end if
18:    end if
19:  end while
20:  return  $-1$ 
21: end function
```

Листинг 4 Псевдокод модифицированного алгоритма Кнута – Морриса – Пратта

На входе: строка $w = a_1 \dots a_n$, подстрока $x = b_1 \dots b_m$, массив суффиксов π , ма

```
1: function KMPорт( $w, x, \pi, badChars$ )
2:    $n \leftarrow length(w)$ 
3:    $m \leftarrow length(x)$ 
4:    $shift \leftarrow 0$ 
5:   while  $shift \leq n - m$  do
6:      $j \leftarrow m - 1$ 
7:     while  $j \geq 0$  and  $x_j = w_{shift+j}$  do
8:        $j \leftarrow m - 1$ 
9:     end while
10:    if  $j < 0$  then return  $shift$ 
11:    else
12:       $shift \leftarrow shift + \max(\pi[j], badChars[text[shift + j]])$ 
13:    end if
14:  end while
15:  return  $-1$ 
16: end function
```

Вывод

В данном разделе были описаны псевдокоды для алгоритма Кнута – Морриса – Пратта и его модификации.

3 Технологический раздел

В данном разделе будут описаны средства реализации программного обеспечения, а также представлены листинги и функциональные тесты.

3.1 Средства реализации

3.2 Сведения о модулях программы

3.3 Реализация алгоритмов

3.4 Тестирование

Вывод

4 Исследовательский раздел

Заключение

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Охотин. А. С.* Математические основы алгоритмов. — 2022. — URL: https://users.math-cs.spbu.ru/~okhotin/teaching/algorithms1_2022 ; Лекция 6.
2. *Ваныкина. Г. В.* Структуры и алгоритмы компьютерной обработки данных. — 2012. — URL: <https://intuit.ru/studies/courses/648/504/info> ; Лекция 40.