**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №4**

**по дисциплине «Построение и анализ алгоритмов»**

Тема: Кнут-Моррис-Пратт

|  |  |  |
| --- | --- | --- |
| Студент гр. 9382 |  | Русинов Д.А. |
| Преподаватель |  | Фирсов М.А. |

Санкт-Петербург

2021

## Цель работы.

Изучить алгоритм Кнута-Морриса-Пратта и создать программу, которая находит все вхождения подстроки в тексте. Также реализовать программу, которая определяет, является ли строка циклическим сдвигом другой строки.

## Задание 1

## Реализуйте алгоритм КМП и с его помощью для заданных шаблона *P* (*∣P∣≤15000*) и текста *T* (*∣T∣≤5000000*) найдите все вхождения *P* в *T*.

Вход:

Первая строка - *P*

Вторая строка - *T*

Выход:

индексы начал вхождений *P* в *T*, разделенных запятой, если *P* не входит в *T*, то вывести *−1*

**Sample Input:**

ab

abab

**Sample Output:**

0,2

**Задание 2**

Заданы две строки *A* (*∣A∣≤5000000*) и *B* (*∣B∣≤5000000*).

Определить, является ли *А* циклическим сдвигом *В*(это значит, что *А* и *В* имеют одинаковую длину и *А* состоит из суффикса *В*, склеенного с префиксом *В*). Например, defabc является циклическим сдвигом abcdef.

Вход:

Первая строка - *A*

Вторая строка - *B*

Выход:

Если *A* вляется циклическим сдвигом *B*, индекс начала строки *B* в *A*, иначе вывести *−1*. Если возможно несколько сдвигов вывести первый индекс.

**Sample Input:**

defabc

abcdef

**Sample Output:**

3

**Описание алгоритма**

Первоначально определяются значения префикс-функции для шаблона, который необходимо найти в тексте. Значение префикс-функции означает длину наибольшего совпадения префикса и суффикса в подстроке шаблона, которая рассматривается. Всего значений будет N, где первое значение – 0, так как размер подстроки равен единице, соответственно максимальный размер префикса и суффикса этой подстроки равен единице.

В первой программе необходимо найти все вхождения шаблона в тексте. Рассматриваются символы текста до тех пор, пока не будет рассмотрен конечный символ. Также рассматриваются символы строки-шаблона. Если символ текста и символ шаблона равен, то рассматриваются следующие символы. Если этот символ был последним символом строки-шаблона, то было найдено вхождение шаблона в тексте, индекс вхождения записывается в результат. Индекс строки-шаблона в этом случае становится значением префикс-функции под предыдущим значением индекса строки-шаблона.

Если рассматриваемые символы не равны, и рассматриваемый символ строки-шаблона был начальным, то сдвигается индекс символа, который рассматривается в тексте на единицу. Если же символ был не начальным, то индекс символа в строке-шаблоне становится равен значению префикс-функции предыдущего индекса.

Во второй программе происходит конкатенация строки, для которой необходимо определить является ли она циклическим сдвигом другой строки, с собой же. Затем в полученной строке происходит поиск изначальной строке с помощью алгоритма КМП. Если удалось найти вхождение, то строка является циклическим сдвигом.

**Оценка сложности по памяти**

В обоих программах необходимо хранить две строки – шаблон и текст, а также нужно хранить вектор с значениями префикс-функции. Сложность составляет O(N + M), где N – длина первой строки, M – длина второй строки.

**Оценка сложности по времени**

Значение префикс-функции вычисляется за O(M) сравнений, где M – длина строки-шаблона, так как необходимо пройтись по всей строке, чтобы определить префикс-функцию.

Поиск строки-шаблона в тексте с помощью алгоритма КМП будет занимать O(N), где N – длина текста, так как весь текст будет пройден ровно один раз. Каждый символ текста будет рассмотрен ровно один раз за счет использования префикс-функции.

Итоговая оценка – O(M + N).

Оценка для второй программы будет такой же.

**Тестирование**

Результаты тестирования программы можно посмотреть в приложениях Б и В.

## Выводы.

Был изучен алгоритм Кнута-Морриса-Пратта и создана программа, которая находит все вхождения подстроки в тексте. Также была реализована программа, которая определяет, является ли строка циклическим сдвигом другой строки.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.cpp

#include "string"

#include "vector"

#include "iostream"

#define task 1

std::string getArrowString(int stringLength, int index) {

auto string = std::string();

for (int i = 0; i < stringLength; ++i) {

if (i == index) string += "\u2193";

else string += " ";

}

return string;

}

void print(std::string & first, std::string & second,

int firstIndex, int secondIndex, const std::string& information) {

std::cout << getArrowString(first.length(), firstIndex) << " | ";

std::cout << getArrowString(second.length(), secondIndex) << '\n';

std::cout << first << " | " << second << '\n';

std::cout << information << "\n\n";

}

std::vector<int> getPrefixFunction(std::string & pattern) {

auto prefixFunction = std::vector<int>(pattern.length());

// вычисляем префикс-функцию для строки-шаблона

int i = 1;

int j = 0;

// индексы символов, который мы в данный момент рассматриваем

// и сравниваем

std::cout << "[Формирование префикс-функции]" << std::endl;

while (i != pattern.length()) {

std::cout << "i = " << std::to\_string(i) << "; j = " << std::to\_string(j) << "\n";

if (pattern[i] == pattern[j]) {

// если символы равны, то запишем в префикс функцию

// для символа i образ j + 1

auto information = std::string("Символы равны, поэтому образ функции под индексом ") + std::to\_string(i) + std::string(" равен ") + std::to\_string(j + 1);

information += ", j и i увеличиваются на 1";

print(pattern, pattern, j, i, information);

prefixFunction[i] = j + 1;

i++;

j++;

} else if (j == 0) {

// если символ j начальный, и символы i j не равны,

// то образ функции для i = 0

auto information = std::string("Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом ") + std::to\_string(i) + std::string(" равен 0");

information += ", i увеличивается на 1";

print(pattern, pattern, j, i, information);

prefixFunction[i] = 0;

i++;

} else {

auto information = std::string("Символы не равны, и символ под индексом j - не начальный, поэтому новое значение j - ") + std::to\_string(prefixFunction[j - 1]);

print(pattern, pattern, j, i, information);

j = prefixFunction[j - 1]; // иначе сдвигаем j назад

}

}

return prefixFunction;

}

std::vector<int> getSubstringIndexes(std::string & text, std::string & pattern) {

auto prefixFunction = getPrefixFunction(pattern);

int textIndex = 0;

int substringIndex = 0;

auto result = std::vector<int>();

std::cout << "[Поиск шаблона в тексте]" << std::endl;

// пока не рассмотрим весь текст

while (textIndex != text.length()) {

std::cout << "textIndex = " << std::to\_string(textIndex) << "; substringIndex = " << std::to\_string(substringIndex) << "\n";

if (text[textIndex] == pattern[substringIndex]) {

// если символы равны, то смотрим следующие символы

auto info = "";

if (substringIndex + 1 == pattern.length()) {

// если индекс конечный для подстроки, то мы нашли заданную подстроку в тексте

// сдвигаемся назад в подстроке

info = "Символы равны, в подстроке не осталось символов для сравнения, поэтому найдено вхождение подстроки в текст";

print(text, pattern, textIndex, substringIndex, info);

result.push\_back(textIndex - substringIndex);

substringIndex = prefixFunction[substringIndex];

} else {

info = "Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы";

print(text, pattern, textIndex, substringIndex, info);

}

textIndex += 1;

substringIndex += 1;

} else { // Если символы не равны

// если символ подстроки первый, то сдвигаем индекс текста

if (substringIndex == 0) {

auto info = "Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста";

print(text, pattern, textIndex, substringIndex, info);

textIndex += 1;

}

// если не первый, то сдвигаемся по значению в префикс-функции

else {

auto info = "Символы не равны, рассматривался не начальный символ подстроки, теперь он равен " + std::to\_string(prefixFunction[substringIndex - 1]);

print(text, pattern, textIndex, substringIndex, info);

substringIndex = prefixFunction[substringIndex - 1];

}

}

}

std::cout << "Все символы текста рассмотрены, завершение работы алгоритма" << std::endl;

return result;

}

void firstTask(std::string & text, std::string & pattern) {

auto result = getSubstringIndexes(text, pattern);

if (!result.empty()) {

for (int i = 0; i < result.size() - 1; ++i) std::cout << result[i] << ',';

std::cout << result[result.size() - 1];

} else std::cout << -1;

}

void secondTask(std::string & text, std::string & pattern) {

if (text.length() != pattern.length()) {

std::cout << -1;

return;

}

text = text + text;

auto result = getSubstringIndexes(text, pattern);

if (!result.empty()) std::cout << result[0];

else std::cout << -1;

}

int main() {

std::string text;

std::string pattern;

std::cin >> text;

std::cin >> pattern;

if (task == 1) {

firstTask(text, pattern);

} else {

secondTask(text, pattern);

}

return 0;

}

**ПРИЛОЖЕНИЕ Б**

**ТЕСТИРОВАНИЕ ПРОГРАММЫ 1**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ababab  ab | [Формирование префикс-функции]  i = 1; j = 0  ↓ | ↓  ab | ab  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 1 равен 0, i увеличивается на 1  [Поиск шаблона в тексте]  textIndex = 0; substringIndex = 0  ↓ | ↓  ababab | ab  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 1; substringIndex = 1  ↓ | ↓  ababab | ab  Символы равны, в подстроке не осталось символов для сравнения, поэтому найдено вхождение подстроки в текст  textIndex = 2; substringIndex = 1  ↓ | ↓  ababab | ab  Символы не равны, рассматривался не начальный символ подстроки, теперь он равен 0  textIndex = 2; substringIndex = 0  ↓ | ↓  ababab | ab  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 3; substringIndex = 1  ↓ | ↓  ababab | ab  Символы равны, в подстроке не осталось символов для сравнения, поэтому найдено вхождение подстроки в текст  textIndex = 4; substringIndex = 1  ↓ | ↓  ababab | ab  Символы не равны, рассматривался не начальный символ подстроки, теперь он равен 0  textIndex = 4; substringIndex = 0  ↓ | ↓  ababab | ab  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 5; substringIndex = 1  ↓ | ↓  ababab | ab  Символы равны, в подстроке не осталось символов для сравнения, поэтому найдено вхождение подстроки в текст  Все символы текста рассмотрены, завершение работы алгоритма  0,2,4 |
| oabcdfe  abcdfeo | [Формирование префикс-функции]  i = 1; j = 0  ↓ | ↓  abcdfeo | abcdfeo  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 1 равен 0, i увеличивается на 1  i = 2; j = 0  ↓ | ↓  abcdfeo | abcdfeo  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 2 равен 0, i увеличивается на 1  i = 3; j = 0  ↓ | ↓  abcdfeo | abcdfeo  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 3 равен 0, i увеличивается на 1  i = 4; j = 0  ↓ | ↓  abcdfeo | abcdfeo  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 4 равен 0, i увеличивается на 1  i = 5; j = 0  ↓ | ↓  abcdfeo | abcdfeo  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 5 равен 0, i увеличивается на 1  i = 6; j = 0  ↓ | ↓  abcdfeo | abcdfeo  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 6 равен 0, i увеличивается на 1  [Поиск шаблона в тексте]  textIndex = 0; substringIndex = 0  ↓ | ↓  oabcdfe | abcdfeo  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 1; substringIndex = 0  ↓ | ↓  oabcdfe | abcdfeo  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 2; substringIndex = 1  ↓ | ↓  oabcdfe | abcdfeo  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 3; substringIndex = 2  ↓ | ↓  oabcdfe | abcdfeo  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 4; substringIndex = 3  ↓ | ↓  oabcdfe | abcdfeo  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 5; substringIndex = 4  ↓ | ↓  oabcdfe | abcdfeo  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 6; substringIndex = 5  ↓ | ↓  oabcdfe | abcdfeo  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  Все символы текста рассмотрены, завершение работы алгоритма  -1 |
| qwerty  qwe | 0 [Формирование префикс-функции]  i = 1; j = 0  ↓ | ↓  qwe | qwe  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 1 равен 0, i увеличивается на 1  i = 2; j = 0  ↓ | ↓  qwe | qwe  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 2 равен 0, i увеличивается на 1  [Поиск шаблона в тексте]  textIndex = 0; substringIndex = 0  ↓ | ↓  qwerty | qwe  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 1; substringIndex = 1  ↓ | ↓  qwerty | qwe  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 2; substringIndex = 2  ↓ | ↓  qwerty | qwe  Символы равны, в подстроке не осталось символов для сравнения, поэтому найдено вхождение подстроки в текст  textIndex = 3; substringIndex = 1  ↓ | ↓  qwerty | qwe  Символы не равны, рассматривался не начальный символ подстроки, теперь он равен 0  textIndex = 3; substringIndex = 0  ↓ | ↓  qwerty | qwe  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 4; substringIndex = 0  ↓ | ↓  qwerty | qwe  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 5; substringIndex = 0  ↓ | ↓  qwerty | qwe  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  Все символы текста рассмотрены, завершение работы алгоритма  0 |
| qwe  qwert | [Формирование префикс-функции]  i = 1; j = 0  ↓ | ↓  qwert | qwert  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 1 равен 0, i увеличивается на 1  i = 2; j = 0  ↓ | ↓  qwert | qwert  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 2 равен 0, i увеличивается на 1  i = 3; j = 0  ↓ | ↓  qwert | qwert  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 3 равен 0, i увеличивается на 1  i = 4; j = 0  ↓ | ↓  qwert | qwert  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 4 равен 0, i увеличивается на 1  [Поиск шаблона в тексте]  textIndex = 0; substringIndex = 0  ↓ | ↓  qwe | qwert  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 1; substringIndex = 1  ↓ | ↓  qwe | qwert  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 2; substringIndex = 2  ↓ | ↓  qwe | qwert  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  Все символы текста рассмотрены, завершение работы алгоритма  -1 |
| sdnvksefljjk  pnkvsnvsklv | [Формирование префикс-функции]  i = 1; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 1 равен 0, i увеличивается на 1  i = 2; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 2 равен 0, i увеличивается на 1  i = 3; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 3 равен 0, i увеличивается на 1  i = 4; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 4 равен 0, i увеличивается на 1  i = 5; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 5 равен 0, i увеличивается на 1  i = 6; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 6 равен 0, i увеличивается на 1  i = 7; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 7 равен 0, i увеличивается на 1  i = 8; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 8 равен 0, i увеличивается на 1  i = 9; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 9 равен 0, i увеличивается на 1  i = 10; j = 0  ↓ | ↓  pnkvsnvsklv | pnkvsnvsklv  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 10 равен 0, i увеличивается на 1  [Поиск шаблона в тексте]  textIndex = 0; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 1; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 2; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 3; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 4; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 5; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 6; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 7; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 8; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 9; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 10; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 11; substringIndex = 0  ↓ | ↓  sdnvksefljjk | pnkvsnvsklv  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  Все символы текста рассмотрены, завершение работы алгоритма  -1 |

**ПРИЛОЖЕНИЕ Г**

**ТЕСТИРОВАНИЕ ПРОГРАММЫ 2**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| abcdef  defabc | abcdef  defabc  [Формирование префикс-функции]  i = 1; j = 0  ↓ | ↓  defabc | defabc  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 1 равен 0, i увеличивается на 1  i = 2; j = 0  ↓ | ↓  defabc | defabc  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 2 равен 0, i увеличивается на 1  i = 3; j = 0  ↓ | ↓  defabc | defabc  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 3 равен 0, i увеличивается на 1  i = 4; j = 0  ↓ | ↓  defabc | defabc  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 4 равен 0, i увеличивается на 1  i = 5; j = 0  ↓ | ↓  defabc | defabc  Символы не равны, и символ под индексом j - начальный, поэтому образ функции под индексом 5 равен 0, i увеличивается на 1  [Поиск шаблона в тексте]  textIndex = 0; substringIndex = 0  ↓ | ↓  abcdefabcdef | defabc  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 1; substringIndex = 0  ↓ | ↓  abcdefabcdef | defabc  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 2; substringIndex = 0  ↓ | ↓  abcdefabcdef | defabc  Символы не равны, рассматривался начальный символ подстроки, поэтому рассматриваем следующий символ текста  textIndex = 3; substringIndex = 0  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 4; substringIndex = 1  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 5; substringIndex = 2  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 6; substringIndex = 3  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 7; substringIndex = 4  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 8; substringIndex = 5  ↓ | ↓  abcdefabcdef | defabc  Символы равны, в подстроке не осталось символов для сравнения, поэтому найдено вхождение подстроки в текст  textIndex = 9; substringIndex = 1  ↓ | ↓  abcdefabcdef | defabc  Символы не равны, рассматривался не начальный символ подстроки, теперь он равен 0  textIndex = 9; substringIndex = 0  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 10; substringIndex = 1  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  textIndex = 11; substringIndex = 2  ↓ | ↓  abcdefabcdef | defabc  Символы равны, и в подстроке еще есть символы для сравнения, рассматриваем следующие символы  Все символы текста рассмотрены, завершение работы алгоритма  3 |
| defabc  abc | -1 |
| aaaaaaaa  bbbbbbbb | -1 |
| abcabcabc  bcabcabca | 1 |
| abcdfeo  oabcdfe | 6 |