**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

по дисциплине «СИАОД»

на тему: «Методы поиска подстроки в строке»

Выполнил: студент группы БВТ1802

Сурин В.И.

Руководитель:

Кутейников Иван Алексеевич

Москва 2020

**Цель работы**: Реализовать заданный метод поиска подстроки в строке в соответствии с индивидуальным заданием. Для всех вариантов добавить реализацию добавления строк, ввода подстроки и поиска подстроки. Предусмотреть возможность существования пробела. Ввести опцию чувствительности / нечувствительности к регистру. Оценить время работы каждого алгоритма поиска и сравнить его со временем работы стандартной функции поиска, используемой в выбранном языке программирования.

**Выполнение работы:**

В качестве стандартного метода поиска использовался метод indexOf. Был реализован алгоритм Кнута-Морриса-Пратта.

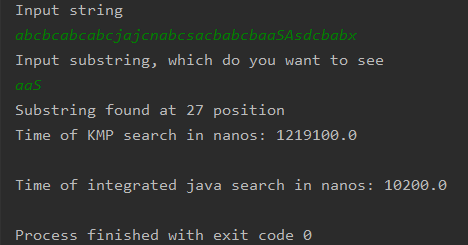
Класс FindSubstring

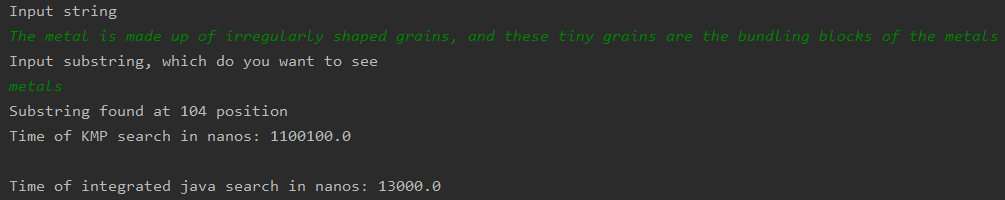
import java.util.Arrays;  
  
public class FindSubstring {  
 public static int[] KMP(char[] pattern, char[] text)  
 {  
 int[] pfl = *pfl*(pattern);  
 int[] indexes = new int[text.length];  
 int size = 0;  
 int k = 0;  
 for (int i = 0; i < text.length; ++i)  
 {  
 while (pattern[k] != text[i] && k > 0)  
 {  
 k = pfl[k - 1];  
 }  
 if (pattern[k] == text[i])  
 {  
 k = k + 1;  
 if (k == pattern.length)  
 {  
 indexes[size] = i + 1 - k;  
 size += 1;  
 k = pfl[k - 1];  
 }  
 }  
 else  
 {  
 k = 0;  
 }  
 }  
 return Arrays.*copyOfRange*(indexes, 0, size);  
 }  
  
 public static int[] pfl(char[] text)  
 {  
 int[] pfl = new int[text.length];  
 pfl[0] = 0;  
  
 for (int i = 1; i < text.length; ++i)  
 {  
 int k = pfl[i - 1];  
 while (text[k] != text[i] && k > 0)  
 {  
 k = pfl[k - 1];  
 }  
 if (text[k] == text[i])  
 {  
 pfl[i] = k + 1;  
 }  
 else  
 {  
 pfl[i] = 0;  
 }  
 }  
  
 return pfl;  
 }  
}

Класс Main

import java.util.Scanner;  
  
public class MainClass {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 char[] text, substring;  
 String string, sub;  
 System.*out*.println("Input string");  
 string = in.nextLine();  
 text = string.toCharArray();  
  
 System.*out*.println("Input substring, which do you want to see");  
 sub = in.nextLine();  
 substring = sub.toCharArray();  
  
 double before = System.*nanoTime*();  
 FindSubstring.*KMP*(substring, text);  
 double after = System.*nanoTime*();  
  
 for (int i = 0; i < FindSubstring.*KMP*(substring, text).length; i++){  
 System.*out*.print("Substring found at " + FindSubstring.*KMP*(substring, text)[i] + " position");  
 }  
  
 System.*out*.println();  
 System.*out*.println("Time of KMP search in nanos: " + (after - before));  
  
 before = System.*nanoTime*();  
 string.indexOf(string);  
 after = System.*nanoTime*();  
  
 System.*out*.println();  
 System.*out*.println("Time of integrated java search in nanos: " + (after - before));  
 }  
}

**Тесты программы:**





|  |  |  |
| --- | --- | --- |
| Тип сортировки | KMP | Java |
| Тест №1 | Строка: abcbcabcabcjajcnabcsacbabcbaaSAsdcbabx  Найти: aaS | |
| Время поиска | 1219100нс | 10200нс |
| Тест №2 | Строка: The metal is made up of irregularly shaped grains, and these tiny grains are the bundling blocks of the metals  Найти: metals | |
| Время поиска | 1100100нс | 13000нс |
| Тест №3 | Строка: a b c d e f g h i j k l m n o p r s t u v w x y z A B C D E F G H I J K L M N O P R S T U V W X Y Z  Найти: A | |
| Время поиска | 1.6209975E9нс | 5000нс |

**Вывод:** в ходе лабораторной работы я изучил алгоритмы поиска подстроки в строке, реализовал алгоритм Кнута-Морриса-Пратта. Сравнил время поиска реализованного метода со встроенным поиском.