**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 №3**

По дисциплине «СиАОД»

На тему:

«Методы поиска подстроки в строке»

Выполнил: студент группы БВТ1801

Алмамма Башар

Руководитель:

Кутейников Иван Алексеевич

Москва 2020

**Цель работы:**   
Изучить основные методы поиска подстроки в строке и сравнить их скорость.

**Задача (Вариант 1):**Реализовать методы поиска:1. Реализовать на языке Java упрощенный метод Бойера-Мура.   
2. Реализовать поиск стандартной библиотекой Java.  
Сравнить все вышеперечисленные методы по времени.

**Программа:**

**1.** Код программы:

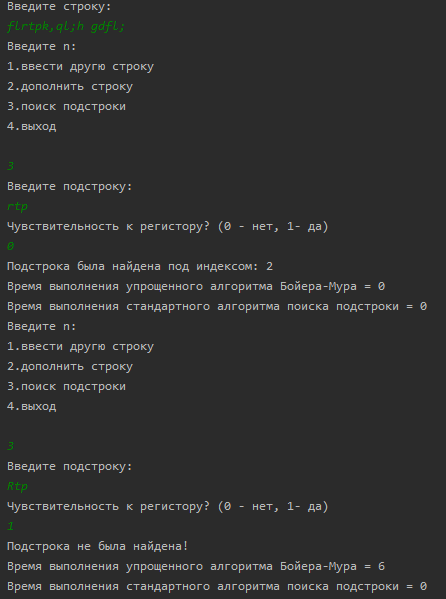
Класс Main:

public class Main {  
  
 public static void main(String[] args) throws IOException {  
 String input, string, string2, substring;  
 int n = 0;  
 long start, end, elapsedTime1, elapsedTime2;  
 BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
 System.out.println("Введите строку:");  
 string= br.readLine();  
 do{  
 System.out.println("Введите n: \n" + "1.ввести другю строку\n" + "2.дополнить строку\n" + "3.поиск подстроки\n" + "4.выход\n");  
 input= br.readLine();  
 n = Integer.parseInt(input);  
 switch (n){  
 case 1:  
 System.out.println("Введите строку:");  
 string= br.readLine();  
 break;  
 case 2:  
 System.out.println("Введите строку:");  
 input= br.readLine();  
 string += input;  
 System.out.println("Новая полученная строка:\n" + string);  
 break;  
 case 3:  
 System.out.println("Введите подстроку:");  
 substring= br.readLine();  
 System.out.println("Чувствительность к регистору? (0 - нет, 1- да)");  
 input = br.readLine();  
 string2 = string;  
  
 if(input.equals("0")){  
 string2 = string.toLowerCase();  
 substring = substring.toLowerCase();  
 }  
  
 start = System.currentTimeMillis();  
 int index = SubstringSearch.Boyer(string2,substring);  
 end = System.currentTimeMillis();  
 elapsedTime1 = end - start;  
  
 start = System.currentTimeMillis();  
 index = string2.indexOf(substring);  
 end = System.currentTimeMillis();  
 elapsedTime2 = end - start;  
  
 if(index != -1){  
 System.out.println("Подстрока была найдена под индексом: "+ index);  
 }  
 else {  
 System.out.println("Подстрока не была найдена!");  
 }  
 System.out.println("Время выполнения упрощенного алгоритма Бойера-Мура = " + elapsedTime1+  
 "\nВремя выполнения стандартного алгоритма поиска подстроки = " + elapsedTime2);  
 break;  
 default:  
 break;  
 }  
 }while(n != 4);  
 }  
}

Класс SubstringSearch:

public class SubstringSearch {  
 public static int Boyer(String, String substring) {  
 int substringLen = substring.length();  
 int stringLen = string.length();  
 if (substringLen > stringLen) {  
 return -1;  
 }  
 HashMap<Character, Integer> t = new HashMap<Character, Integer>();  
 for (int i = 0; i <= 255; i++) {  
 t.put((char) i, substringLen);  
 }  
 for (int i = 0; i < substringLen - 1; i++) {  
 t.put(substring.charAt(i), substringLen - i - 1);  
 }  
 int i = substringLen - 1;  
 int j = i;  
 int k = i;  
 while (j >= 0 && i <= stringLen - 1) {  
 j = substringLen - 1;  
 k = i;  
 while (j >= 0 && string.charAt(k) == substring.charAt(j)) {  
 k--;  
 j--;  
 }  
 i += t.get(string.charAt(i));  
 }  
 if (k >= stringLen - substringLen) {  
 return -1;  
 } else {  
 return k + 1;  
 }  
 }  
}

**2.** Тестирование программы



**4.** Для реализации поиска стандартными средствами java был выбран поиск indexOf().

Сравнение времени выполнения(скорости) методов:

|  |  |  |  |
| --- | --- | --- | --- |
| **Длина текста** | **Длина подстроки** | **Упрощенный метод Бойера-Мура** | **Поиск Java** |
| 1000 | 6 | 0 | 0 |
| 1000 | 15 | 0 | 0 |
| 10000 | 11 | 2 | 0 |
| 100000 | 24 | 4 | 6 |
| 100000 | 14 | 0 | 5 |

**Вывод:** из вышеперечисленных данных можно сделать вывод, что упрощенный метод Бойера-Мура не уступает стандартному алгоритму java по скорости.