

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Поиск образа в строке
Тема

Преподаватель

Р. Ю. Царев

подпись, дата

инициалы, фамилия

Студент

КИ19-17/16 031939175

номер группы, зачетной
книжки

подпись, дата

А. Д. Непомнящий

инициалы, фамилия

Красноярск 2020

1 Цель работы

Изучение и осознание некоторых алгоритмов поиска образа в строке.

2 Задачи

Написать программу поиска образа в строке по методу Боуера и Мура. Предусмотреть возможность существования в образе пробела. Ввести опцию чувствительности или нечувствительности к регистру.

Предъявлены следующие требования к выполнению работы.

1. Строгое соответствие программы и результатов ее работы с полученным заданием.
2. Самостоятельное тестирование и отладка программы.
3. Устойчивость работы программы при любых воздействиях, задаваемых пользователем через интерфейс программы.
4. Предоставление демонстрационного примера и исходного текста программы для защиты.
5. Предоставление отчета по практическому заданию, содержащего описание реализованного алгоритма, программы, результатов работы программы (отчет необходимо загрузить на сайт курса).

3 Описание реализованного алгоритма

Реализован алгоритм Боуера и Мура.

4 Описание программы

Для решения задачи были написаны модуль `boyer_moore_horspool.py`, содержащий функцию, реализующую алгоритм Боуера и Мура и модуль `main.py`, реализующий пользовательский интерфейс. Ниже приведены листинги кода.

Листинг 1 – Код в файле boyer_moore_horspool.py

```
def find_substring(origin_string, origin_substring, is_case_sensitive):
    shift = dict()
    obj_string = origin_string[
        : ] if is_case_sensitive else origin_string.casefold()
    obj_substring = origin_substring[
        : ] if is_case_sensitive else origin_substring.casefold()

    for i in range(len(obj_substring)):
        index = ((-i - 2) % (len(obj_substring)))
        if obj_substring[index] not in shift.keys():
            shift[obj_substring[index]] = i + 1

    position = 0
    while position < len(obj_string) - len(obj_substring) + 1:
        for i in reversed(range(len(obj_substring))):
            if obj_substring[i] != obj_string[position + i]:
                position += shift.get(
                    obj_string[position + len(obj_substring) - 1],
                    len(obj_substring))
                break
        return position
    return None
```

Листинг 2 – Код в файле main.py

```
import boyer_moore_horspool

if __name__ == '__main__':
    string = input("Enter origin string\n")
    substring = input("Enter a substring to find\n")
    casesens = True if input(
        "Enter \"Y\" if case sensitivity required,"
        "otherwise enter anything else\n") == 'Y' else False
    print(boyer_moore_horspool.find_substring(string, substring, casesens))
```

5 Результаты работы программы

На следующем рисунке приведен скриншот с результатами работы программы.

```
PS G:\Projects\AlgorithmsAndDataStructures\Lab1> python main.py
Enter origin string
Find this substring
Enter a substring to find
su
Enter "Y" if case sensitivity required,otherwise enter anything else
10
PS G:\Projects\AlgorithmsAndDataStructures\Lab1> python main.py
Enter origin string
Find this substring
Enter a substring to find
s su
Enter "Y" if case sensitivity required,otherwise enter anything else
8
PS G:\Projects\AlgorithmsAndDataStructures\Lab1> python main.py
Enter origin string
Find this subString
Enter a substring to find
S
Enter "Y" if case sensitivity required,otherwise enter anything else
Y
13
PS G:\Projects\AlgorithmsAndDataStructures\Lab1> python main.py
Enter origin string
Find this subString
Enter a substring to find
S
Enter "Y" if case sensitivity required,otherwise enter anything else
No
8
```

Рисунок 1 – Результаты работы программы