МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

АНАЛИЗ АЛГОРИТМА БОЙЕРА — МУРА

ОТЧЕТ

студентки 2 курса 211 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета компьютерных наук и информационных технологий
Никитенко Яны Валерьевны

Саратов 2025

СОДЕРЖАНИЕ

1 Текст программы

```
    // Функция для создания таблицы смещений по символам
vector<int> CreateBadCharTable(const string& pattern) {
    const int ALPHABET_SIZE = 256;
    vector<int> badCharTable(ALPHABET_SIZE, -1);

    for (int i = 0; i < pattern.size(); ++i) {
        badCharTable[static_cast<int>(pattern[i])] = i;
    }

    return badCharTable;
}
//

// Функция для поиска подстроки с использованием алгоритма Бойера — Мура
void BoyerMoore(const string& text, const string& pattern) {
    int m = pattern.size();
    int n = text.size();
    vector<int> badCharTable = CreateBadCharTable(pattern);

    int s = 0;
    while (s <= n - m) {
        int j = m - 1;

        while (j >= 0 && pattern[j] == text[s + j]) {
            j--;
        }

        if (j < 0) {
            cout << s << endl;
          s += (s + m < n) ? m - badCharTable[static_cast<int>(text[s + m])] : 1;
        }
        else {
            s += max(1, j - badCharTable[static_cast<int>(text[s + j])]);
```

```
        }
    }
}
//
```

2 Анализ

Сдвиги

Внутри основного цикла может быть выполнено не более n/m сдвигов (каждый сдвиг хотя бы на одну позицию). Кроме того, при каждом сдвиге мы можем использовать информацию из таблицы "плохих символов"для определения дополнительного сдвига. В худшем случае для каждого сдвига может потребоваться O(m) времени, чтобы определить этот сдвиг.

Таким образом, временная сложность алгоритма Бойера-Мура в худшем случае составляет: O(m)+O(n) * O(m) = O(nm)

Это означает, что в худшем случае алгоритм будет выполняться за время, пропорциональное произведению длины текста на длину шаблона.

В лучшем случае алгоритм использует преимущества таблицы "плохих символов"для значительных сдвигов, что уменьшает количество сравнений и сдвигов:

При каждом сдвиге шаблон смещается на значительное расстояние, m позиций сразу, если символ, следующий за совпадением, отсутствует в шаблоне. Количество сдвигов ограничено n / m. В лучшем случае O(m)+O(n/m)*O(m)=O(m)+O(n)=O(n) O(m) — время на построение таблицы "плохих символов". O(n/m) — количество сдвигов шаблона по тексту. O(m) — время на проверку символов шаблона на каждой позиции (но в лучшем случае это практически всегда константное время, так как шаблон сразу сдвигается на m позиций).