

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій
Кафедра програмного забезпечення



ЗВІТ

до лабораторної роботи №12

на тему: «Алгоритм Бойера-Мура для пошуку у стрічці.»

з дисципліни: «Алгоритми і структури даних»

Лектор:

доц. кафедри ПЗ

Коротєєва Т. О.

Виконав:

ст. гр. ПЗ-22

Чаус О. М.

Прийняв:

асист. кафедри ПЗ

Франко А. В.

« ____ » _____ 2022 р.

Σ = ____

Тема роботи: Алгоритм Бойера-Мура для пошуку у стрічці.

Мета роботи: навчитися застосовувати алгоритм Бойера для пошуку у стрічці при розв'язанні задач та перевірити його ефективність на різних масивах даних. Експериментально визначити складність алгоритму.

Теоретичні відомості

Алгоритм пошуку рядка Бойера — Мура, - ефективний алгоритм пошуку рядка, який є еталоном при практичних дослідженнях алгоритмів пошуку рядка. Був розроблений Робертом Бойером і Джеєм Муром у 1977 році. Перевага цього алгоритму в тому, що ціною деякої кількості попередніх обчислень над зразком або шаблоном (але не над рядком, в якому ведеться пошук) шаблон порівнюється з вихідним текстом не у всіх позиціях — частина перевірок пропускаються як такі, що не дадуть результату. Цей алгоритм швидко працює у ситуаціях коли зразок набагато коротший від тексту пошуку, або коли відбувається пошук в декількох документах. Зазвичай, чим довше зразок, тим швидше працює алгоритм.

Індивідуальне завдання

Дано текст що складається з n стрічок. Визначити чи є серед даного тексту цифри, якщо є вилучити їх з тексту. Знайти задане слово в тексті.

Код програми

```
void remove_digits(std::string &str) {
    str.erase(std::remove_if(str.begin(), str.end(), [](char a) { return
isdigit(a); }), str.end());
}

std::vector<std::size_t> bm_search(const std::string txt, const std::string
wrd){
    std::vector<std::size_t> res;
    const std::size_t txt_size {txt.size()};
    const std::size_t wrd_size {wrd.size()};

    std::size_t shift {0};

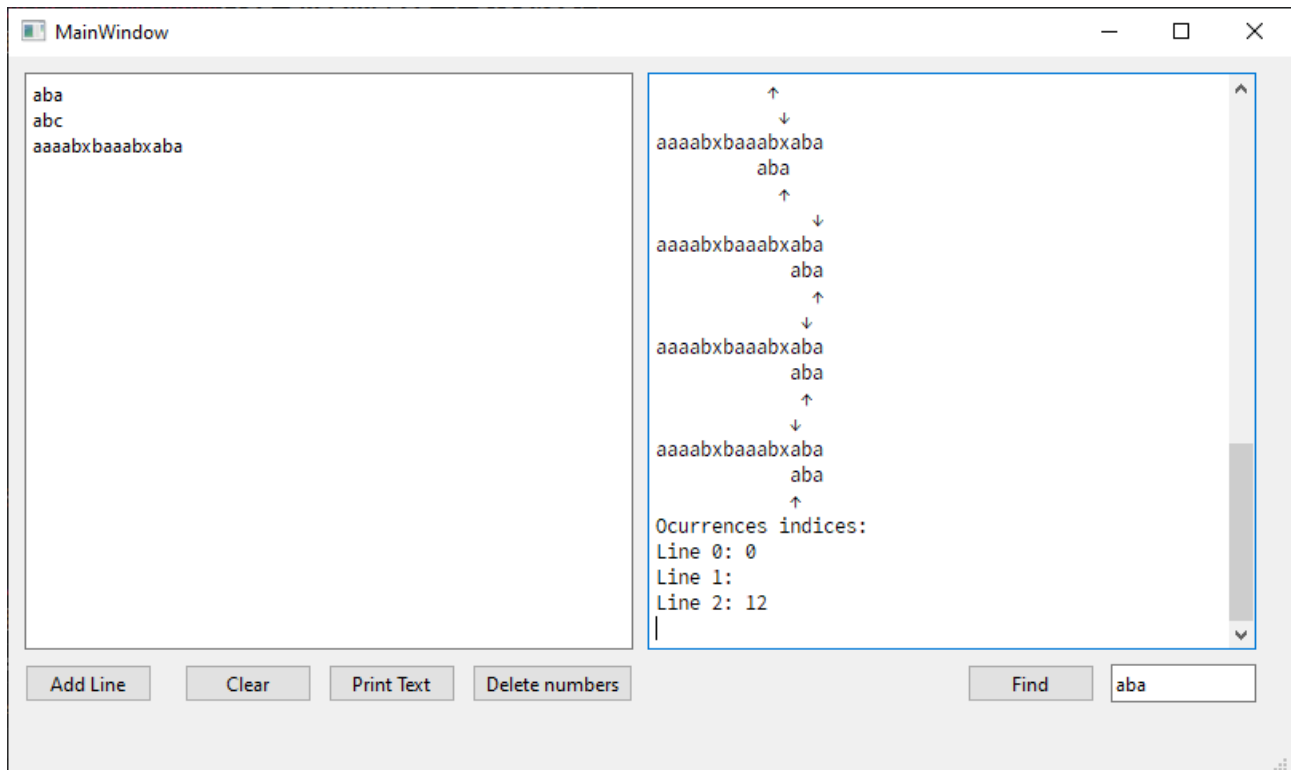
    while (shift <= (txt_size - wrd_size)) {
        int j {static_cast<int>(wrd_size - 1)};    //j points to last char in
word
        while(j >= 0) { // check if pattern occurs at this shift
            if(wrd[j] != txt[shift + j])
                break;
            j--;
        }
        if (!(j < 0)) { //there is a mismatch
            const char bad_char = txt[shift + j];
            int bc_idx {-1} ; //look for mismatched character in pattern
            for (std::size_t i = wrd_size; i >= 0; i--) {
                if(wrd[i] == bad_char) {
                    bc_idx = i;
                    break;
                }
            }
            if(bc_idx > static_cast<int>(j)) //if mismatched character in
pattern comes after current character
                shift++;
            else if(bc_idx >= 0) //else if there is such character in pattern
                shift += (j - bc_idx);
        }
    }
}
```

```

        else //else if there is no such character in pattern
            shift += j + 1;
    }
    else {
        res.push_back(shift);
        shift += 1;
    }
}
return res;
}

```

Зображення програми



Висновок:

навчився застосовувати алгоритм Бойера для пошуку у стрічці при розв'язанні задач та перевірів його ефективність на різних масивах даних. Експериментально визначити складність алгоритму $O(n + m)$