

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 12
На тему: "Алгоритм пошуку Бойера Мура"
З дисципліни: "Алгоритми та структури даних"

Лектор:
доцент кафедри ПЗ
Коротєєва Т.О.

Виконав:
студент групи ПЗ-22
Коваленко Д.М.

Прийняв:
асистент кафедри ПЗ
Франко А.В.

«_____» _____ 2022 р.
 Σ = _____

Тема. Алгоритм пошуку Бойера Мура.

Мета. Навчитися застосовувати алгоритм пошуку Бойера Мура при розв'язуванні задач та перевірити його ефективність на різних масивах даних. Експериментально визначити складність алгоритму.

Лабораторне завдання

Розробити програму, яка:

3. У файлі задано дві стрічки S та P. Довжина стрічок більша 0 і менша 10000, стрічки містять лише букви латинського алфавіту. Вивести номери символів в порядку спадання починаючи з яких стрічка P входить в стрічку S.

Теоретичні відомості

Нажаль співпадіння зустрічаються значно рідше, ніж неспівпадіння. Тому виграш від використання алгоритму КМП в більшості випадків незначний. Інший алгоритм Бойера-Мура базується на наступній схемі: порівняння символів починається з кінця взірця, а не з початку. Нехай для кожного символу x взірця dx - відстань від самого правого у взірці входження x до кінця взірця. Припустимо, знайдено неспівпадіння між взірцем та текстом. Тоді взірець можна зразу посунути вправо на dx позицій що є більше або рівне 1. Якщо x у взірці взагалі не зустрічається, то посунути взірець можна зразу на всю його довжину m.

В даному алгоритмі розглядається поняття стоп-символа - це є символ в тексті, який є першим неспівпадінням тексту і взірця при порівнянні справа (з кінця взірця). Розглянемо три можливих ситуації:

1. 1. Стоп-символ у взірці взагалі не зустрічається, тоді зсув дорівнює довжині взірця m.

2. 2. Крайня права позиція k входження стоп-символа у взірці є меншою від його позиції j у тексті. Тоді взірець можна зсунути вправо на k-j позицій так, щоб стоп-символ у взірці і тексті опинились один під одним.

3. 3. Крайня права позиція k входження стоп-символа у взірці є більшою від його позиції j у тексті. Тоді зсув дорівнює 1.

Хід роботи

```
fn boyer_moore_search(  
    pattern: &str ,  
    bm: BoyerMoore ,  
    text: &str ,  
    sleep_time: f32 ,  
    stdout: &mut StandardStream ,  
    skip: bool ,  
) -> Result<(Vec<usize> , i32 , i32)> {  
    let mut occurrences = Vec::new();  
    let mut alignments = 0;  
    let mut comparisons = 0;  
  
    let pattern = pattern.as_bytes();  
    let text = text.as_bytes();  
    let mut i = 0;  
  
    while i < text.len() - pattern.len() + 1 {  
        let mut shift = 1;  
        let mut mismatched = false;  
        let mut mismatch_index = 0;  
        let mut skip_bc = 0;  
        let mut skip_gs = 0;  
        alignments += 1;  
  
        for j in (0..pattern.len()).rev() {  
            comparisons += 1;
```

```

        if pattern[j] != text[i + j] {
            skip_bc = bm.bad_char_rule(j, text[i + j] as char)?;
            skip_gs = bm.good_suffix_rule(j)?;
            shift = *[shift, skip_bc, skip_gs].iter().max().unwrap();
            mismatched = true;
            mismatch_index = j;
            break;
        }
    }

    if !mismatched {
        occurrences.push(i);
        skip_gs = bm.match_skip();
        shift = *[shift, skip_gs].iter().max().unwrap();
    }
    if !skip {
        visualize(
            stdout,
            from_utf8(text)?,
            from_utf8(pattern)?,
            i,
            mismatch_index,
            !mismatched,
            sleep_time,
        )?;
        println!("Comparisons: {}", comparisons);

        if i < text.len() - pattern.len() {
            if shift > 0 {
                println!("Shift: {}", shift);
            }

            print!("Press Enter to continue ...\r");
            io.stdout()
                .flush()
                .with_context(|| "Failed to flush stdout")?;
            let mut _input = String::new();
            io.stdin()
                .read_line(&mut _input)
                .with_context(|| "Failed to read input from stdin")?;
            print!("\u{1B}[F\u{1B}[K");
            io.stdout()
                .flush()
                .with_context(|| "Failed to flush stdout")?;
        }

        println!();
    }
    i += shift;
}

Ok((occurrences, alignments, comparisons))
}

```

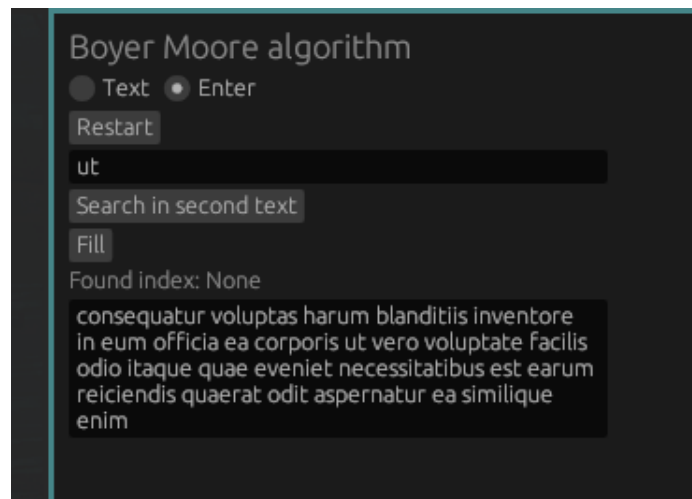


Рис. 1: Вигляд програми

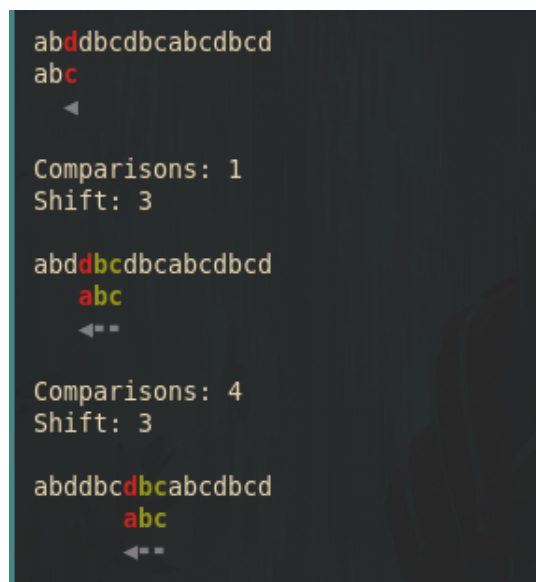


Рис. 2: Покроковий вивід

Висновки

Під час виконання лабораторної роботи я навчився застосовувати алгоритм пошуку Бойєра Мура при розв'язуванні задач та перевірити його ефективність на різних масивах даних. Експериментально визначив складність алгоритму.