Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Алгоритми та складність

Завдання №8

“ Відстань Левенштейна-Дамерау ”

Варіант №2

Виконав студент 2-го курсу

Групи ІПС-21

Лашко Богдан

Київ - 2024

**Завдання**:

Обчислити відстань Левенштейна-Дамерау. Вивести на екран послідовність дій для перетворення першого рядка в другий.

**Теорія**

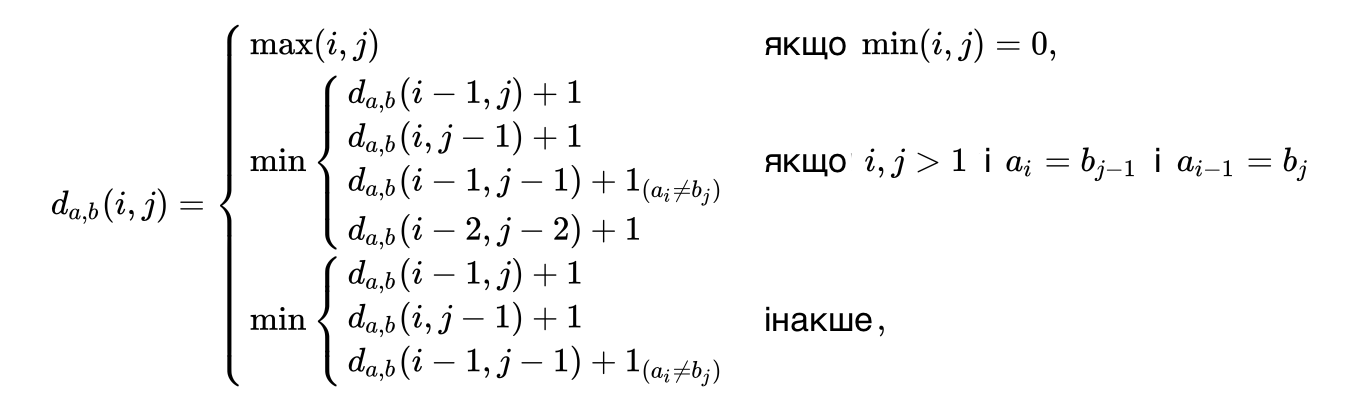
У теорії інформації та комп'ютерних наук, Дамерау-Левенштейн відстань (названа на честь Фредеріка Дамерау та Володимира Левенштейна) - це метрика рядка для вимірювання відстані між двома послідовностями. Простіше кажучи, вона визначає мінімальну кількість операцій (вставок, вилучень, замін або транспозицій одного символу або перестановок двох сусідніх символів), необхідних для перетворення одного слова на інше.

Дамерау-Левенштейн відстань відрізняється від класичної відстані Левенштейна тим, що вона дозволяє також транспозицію (перестановку) символів, в додаток до трьох класичних операцій редагування одного символу (вставка, вилучення та заміна).

У своїй важливій статті, Дамерау вказав, що понад 80% помилок при написанні для системи інформаційного пошуку виникають через одну помилку з чотирьох можливих типів. Він розглядав тільки помилки у написанні, які можна було виправити за допомогою найбільше однієї операції редагування. Хоча первинна мета полягала в вимірюванні відстані між людськими описками для поліпшення програм, таких як перевірка правопису, Дамерау-Левенштейн відстань також застосовується в біології для вимірювання варіацій між послідовностями білків.

**Алгоритм**

Відстань Дамерау — Левенштейна між двома рядками a і b визначаеться функції dab(|a|, |b|) як:



Де **1**(a!=b) індикаторна функція, що дорівнює 0 якщо ai = bj і 1 інакше.

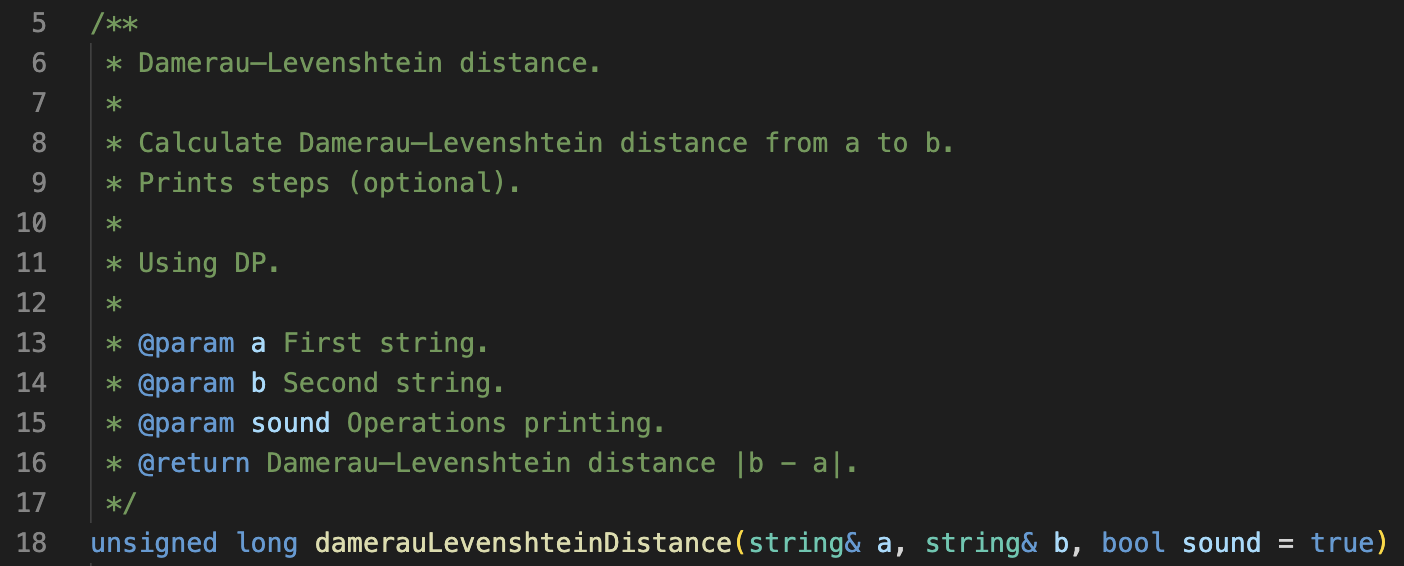
* dab(i – 1, j) + 1 відповідає видаленню символа
* dab(i, j – 1) + 1 відповідає вставці символа
* dab(i – 1, j – 1) + **1**(a!=b) заміна якщо символи різні
* dab(i – 2, j – 2) + 1 перестановка (транспозиція)

**Складність алгоритму**

Складність алгоритму визначення відстані Дамерау-Левенштейна, позначена як O(n \* m), визначається добутком довжин вхідних рядків, де 'n' - це довжина першого рядка, а 'm' - довжина другого рядка. Ця складність виникає з огляду всіх пар символів у вхідних рядках та врахування різних операцій редагування, таких як вставка, видалення, заміна та транспозиція. Алгоритм використовує техніку динамічного програмування, будуючи матрицю для визначення мінімальної кількості редагувань. Хоча алгоритм Дамерау-Левенштейна зазвичай є ефективним для коротких і помірно довгих рядків, його продуктивність може зменшитися для дуже довгих рядків через квадратичний характер часової складності. Існують оптимізаційні техніки для розв'язання конкретних випадків та зменшення константних факторів, але основна складність залишається O(n \* m).

**Мова реалізації алгоритму** С++

**Модулі програми:**



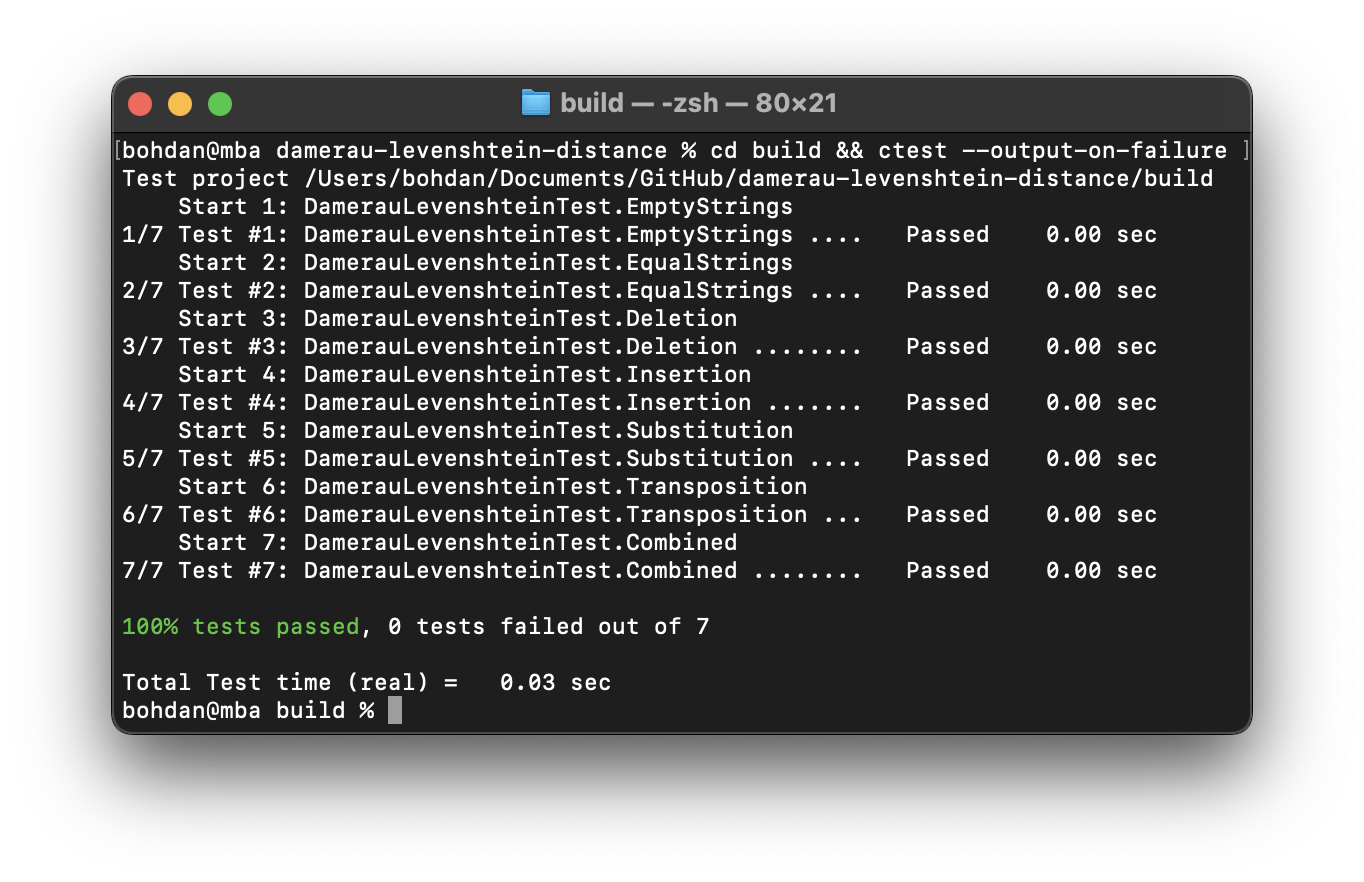
* **unsigned long damerauLevenshteinDistance(string& a, string& b, bool sound = true)**

Функція знаходить відстать між рядками та виводить операції на екран. Складність O(n \* m)

**Інтерфейс користувача**

Введення даних відбувається через консоль. Користувач по черзі вводить два рядки. Програма виводить операції над рядками та шукану відстань.

**Тестові приклади**

Програмно реалізовано тести для кожної операції та деяких крайніх випадків, а також прикладу відстані з різними операціями над рядками (останній тест). (за допомогою googletest) Розглянемо його:

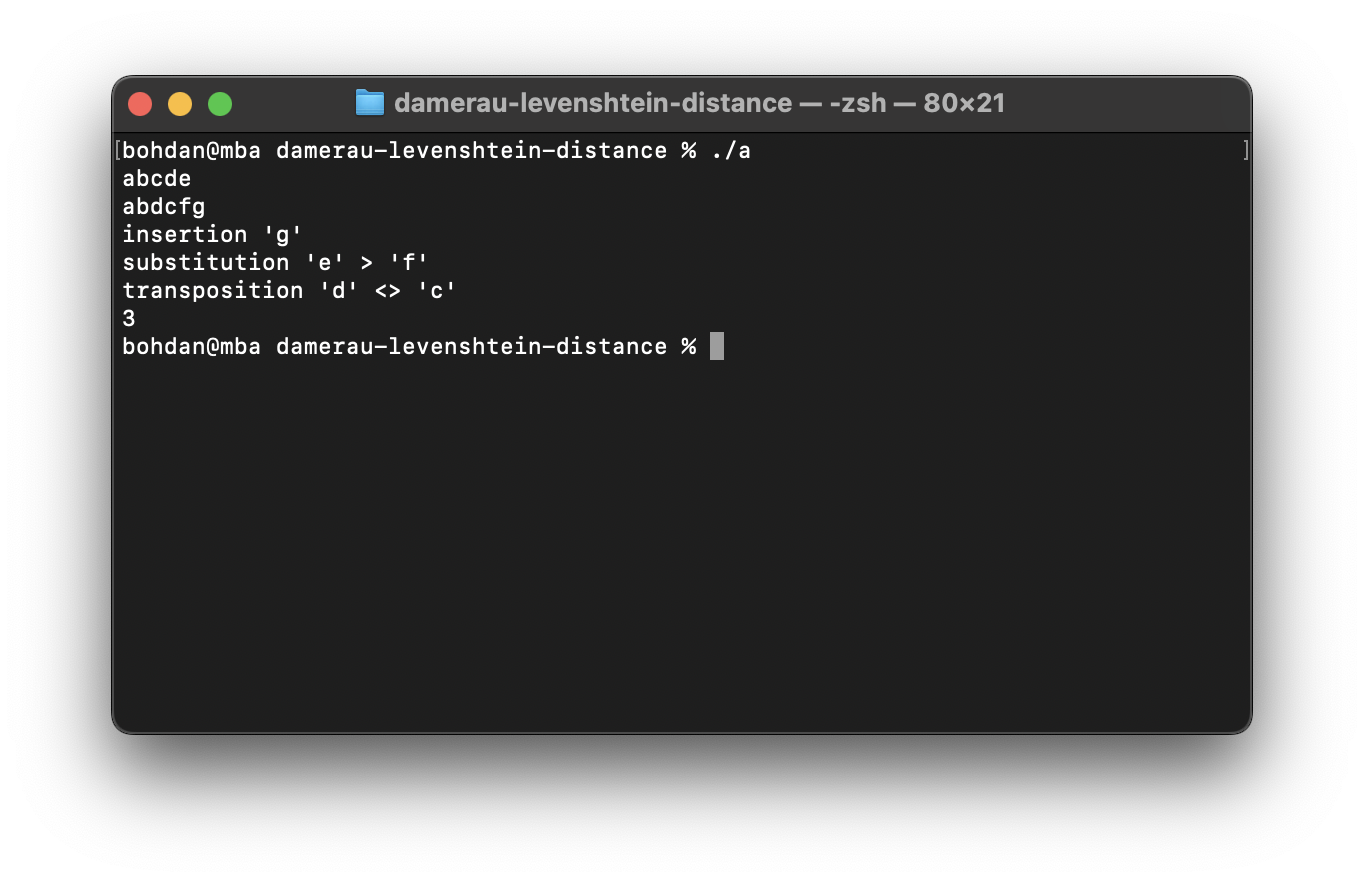
string a = "abcde";

string b = "abdcfg";

відстань має дорівнювати 3, а операції складатися з наприклад (можливі варіації):

вставки g, заміни e на f, транспозиції cd

перевіряємо:



Одержали правильний результат.

**Висновки**

Отже, у даній роботі, було описано алгоритм знаходження відстані Дамерау-Левенштейна. Також було написано програму його реалізації, що використовує метод динамічного програмування, знаходить відстань Дамерау-Левенштейна та виводить операції над рядками.

Так як в моїй реалізації алгоритму Дамерау-Левенштейна динамічне програмування, то складність алгоритму склала O(n\*m) де n та m відповідно довжини рядків.

**Використані літературні джерела**

* «Алгоритми та складність». Лекція №10
* [https://en.wikipedia.org/wiki/Damerau–Levenshtein\_distance](https://en.wikipedia.org/wiki/Damerau–Levenshtein_distance%20)
* [https://google.github.io/googletest/](https://google.github.io/googletest/%20)