*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ*

*КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ*

*ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО*

*НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ*

*ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ*

*КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ*

*НАВЧАЛЬНА ДИСЦИПЛІНА*

*Алгоритми та методи обчислень*

*ЗВІТ*

*З ПРАКТИЧНИХ РОБІТ*

*Виконав:*

*студент групи КН-23-1*

*Гур’єв Д.П*

*Кременчук 2024*

*Практична робота № 7*

Тема. Алгоритми на рядках

Мета: набути практичних навичок застосування базових алгоритмів на рядках та оцінювання їх асимптотичної складності.

***Практична частина 8-варіант***

1.Маємо дві короткі послідовності символів: «HELLO» і «WORLD». Знайти найдовшу спільну підпослідовності символів, використовуючи алгоритм Хаббарда

H E L L O

0 0 0 0 0 0

W 0 0 0 0 0 0

O 0 0 0 0 0 1

R 0 0 0 0 0 1

L 0 0 1 1 1 1

D 0 0 1 1 1 1

Найдовша спільна підпослідовність - “LO”.

Найдовша спільна підпослідовність між “HELLO” і “WORLD” - “LO”

Контрольні запитання

1. У чому полягає задача знаходження найдовшої спільної підпослідовності (LCS)?

Задача знаходження найдовшої спільної підпослідовності (LCS) полягає у визначенні найдовшої підпослідовності, яка спільна для двох (або більше) послідовностей і зберігає порядок символів в обох послідовностях. Це означає, що символи LCS повинні йти в тому ж порядку, в якому вони зустрічаються у вихідних послідовностях, але не обов'язково підряд.

2. Які головні методи можна використовувати для знаходження найдовшої спільної підпослідовності?

Основні методи для знаходження LCS включають:

Динамічне програмування: Найбільш поширений метод, який використовує таблицю для зберігання довжин LCS для різних підпослідовностей і будує рішення поступово.

Рекурсія з мемоізацією: Це вдосконалена версія рекурсивного підходу, яка зберігає результати проміжних обчислень, щоб уникнути повторних обчислень.

Метод довжини матриці: Використовує матрицю для зберігання інформації про збіги символів і їх позицій.

Жадібні алгоритми: Хоча зазвичай не використовуються для LCS через їх неефективність, вони можуть бути корисні в специфічних випадках або як частина складніших алгоритмів.

3. Як працює алгоритм динамічного програмування для знаходження LCS?

Алгоритм динамічного програмування для знаходження LCS включає наступні кроки:

Ініціалізація таблиці: Створюється двовимірна таблиця розміром (m+1)×(n+1), де 𝑚m і 𝑛n — довжини двох послідовностей.

Заповнення таблиці: Для кожного символу обох послідовностей перевіряється, чи збігаються вони:

Якщо символи збігаються, значення в поточній комірці таблиці дорівнює значенню у верхньо-лівій діагональній комірці плюс один.

Якщо символи не збігаються, значення дорівнює максимальному значенню між верхньою і лівою комірками.

Відновлення LCS: Після заповнення таблиці довжина LCS знаходиться в нижньому правому куті таблиці. Підпослідовність можна відновити, рухаючись назад від цієї точки і вибираючи символи, які збігаються.

4. Як працює алгоритм Хаббарда для знаходження LCS?

Алгоритм Хаббарда є одним з методів для знаходження LCS, і він базується на динамічному програмуванні. Загалом, він схожий на базовий алгоритм динамічного програмування, але може включати певні оптимізації або зміни для покращення ефективності в конкретних випадках. Основні кроки включають:

Ініціалізація: Створення і ініціалізація таблиці.

Заповнення таблиці: Порівняння символів і заповнення таблиці відповідно до збігів та оптимального вибору з попередніх комірок.

Відновлення LCS: Відновлення найдовшої спільної підпослідовності з таблиці.

Конкретні оптимізації можуть варіюватися, але загалом алгоритм Хаббарда буде подібним до алгоритму динамічного програмування.

5. Які переваги та недоліки алгоритмів динамічного програмування та Хаббарда для знаходження LCS?

Динамічне програмування:

Переваги:

Гарантує знаходження оптимального рішення.

Добре зрозумілий і широко використовуваний.

Недоліки:

Вимагає O(m×n) пам'яті для таблиці.

Може бути повільним для дуже великих послідовностей через обчислювальну складність O(m×n).

Алгоритм Хаббарда:

Переваги:

Може включати оптимізації, які знижують пам'яттєві або обчислювальні вимоги.

Можливі покращення ефективності у специфічних випадках.

Недоліки:

Складніший для реалізації.

Не завжди гарантує суттєві покращення порівняно з базовим методом динамічного програмування.

6. Які існують практичні застосування для задачі знаходження найдовшої спільної підпослідовності?

Задача LCS має багато практичних застосувань, включаючи:

Порівняння текстів: Використовується в текстових редакторах та системах керування версіями для визначення різниць між документами.

Біоінформатика: Використовується для порівняння геномних послідовностей та знаходження спільних генетичних структур.

Системи пошуку та витягу інформації: Допомагає у виявленні схожих документів або фрагментів тексту.

Обробка природної мови (NLP): Використовується для задач на зразок перевірки граматики, порівняння речень, виявлення плагіату.

Кодування і передачі даних: Допомагає в задачах стискання даних та виявлення помилок при передачі.