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

Ярковий Т.С.

Кременчук 2023

**Практична робота № 7**

**Тема. Алгоритми на рядках**

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

**Завдання**

5. Маємо дві короткі послідовності символів: «XYBAC» і «ABXC». Знайти найдовшу спільну підпослідовності символів, використовуючи алгоритм динамічного програмування.

s1 = «XYBAC»

s2 = «ABXC»

Будуємо таблицю, що на початку заповнена нулями

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | A | B | X | C |
|  | 0 | 0 | 0 | 0 | 0 |
| X | 0 | 0 | 0 | 0 | 0 |
| Y | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 0 | 0 |

Обчислюємо значення.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | A | B | X | C |
|  | 0 | 0 | 0 | 0 | 0 |
| X | 0 | 0 | 0 | 1 | 1 |
| Y | 0 | 0 | 0 | 1 | 1 |
| B | 0 | 0 | 1 | 1 | 1 |
| A | 0 | 1 | 1 | 1 | 1 |
| C | 0 | 1 | 1 | 1 | 2 |

Найдовша підпослідовність має довжину 2.

Далі відновлюємо саму послідовність.

Початкові значення i=5, j=4

Починаємо з останньої комірки dp[5][4], яка має значення 2. Символ s1[4] = s2[3], Додаємо символ “C”. i-=1; j-=1; Зараз значення: i=4, j=3

Порівнюємо s1[3] і s2[2], вони різні, отже перевіряємо умову dp[i - 1][j] > dp[i][j - 1]. dp[3][3] > dp[4][2];

1 > 1, результат false, отже j-=1; Зараз значення: i=4, j=2.

Порівнюємо s1[3] і s2[1], вони різні, отже перевіряємо умову dp[i - 1][j] > dp[i][j - 1]. dp[3][2] > dp[4][1];

1 > 1, результат false, отже j-=1; Зараз значення: i=4, j=1.

Порівнюємо s1[3] і s2[0], вони співпадають. Додаємо символ “A”. i-=1; j-=1; Зараз значення: i=3, j=-1.

j<0, Отже цикл завершується, отримана послідовність “AC” з довжиною 2.

**Контрольні питання**

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

Задача знаходження найдовшої спільної послідовності (підпослідовності) (LCS) полягає в пошуку найбільшої за довжиною послідовності, яка є підпослідовністю двох або більше заданих рядків.

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

Динамічне програмування, рекурсивний алгоритм з використанням мемоізації, алгоритм Хаббарда, алгоритм повного перебору.

3. Як працює алгоритм динамічного програмування для знаходження LCS?

Використовує таблиця для зберігання проміжних результатів. Заповнюючи таблицю ми отримуємо довжину найбільшої підпослідовності, і потім отримуємо саму підпослідовність.

4. Як працює алгоритм Хаббарда для знаходження LCS?

Він подібний до динамічного програмування, але у ньому використовується рекусивний підхід: задача поділяється на 2 підзадачі, і якщо можливо, то для кожної підзадачі йде таке ділення далі.

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

Основними перевагами цих алгоритмів є їхня складність P(m\*n), що набагато швидше за алгоримт повного перебору.

Мінусом обох алгоритмів є складність їх використання для пошуку в великих рядках, через побудову і заповнення таблиці.

Перевагою динамічного програмування є більша просота реальзації.

А перевагою алгоритму Хаббарда є використання рекурсивного підходу, що дозволяє зменшити кількість необхідних операцій. Але через це зростає складність його реалізації.

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

Порівняння текстів, розв’язання задач на графах і біологічні застосування, такі як порівняння генетичних послідовностей.