МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ІМЕНІ МИХАЙЛА
ОСТРОГРАДСЬКОГО

Кафедра комп’ютерної інженерії та електроніки

ЗВІТ З ПРАКТИЧНОЇ РОБОТИ №7

з навчальної дисципліни

«Алгоритми та структури даних**»**

Тема «Алгоритми на рядках»

Студентка гр. КН-24-1 Бояринцова П. С.
Викладач Сидоренко В. М.

Кременчук 2025

Тема роботи: Алгоритми на рядках

1.1 Постановка завдання

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

Завдання: знайти найдовшу спільну підпослідовність**.**

1.2 Розв’язання задачі

Завдання

3. Маємо дві короткі послідовності символів: «ABCDF» і «ACEDB». Знайти найдовшу спільну підпослідовності символів, використовуючи алгоритм динамічного програмування.

Розв’язання.

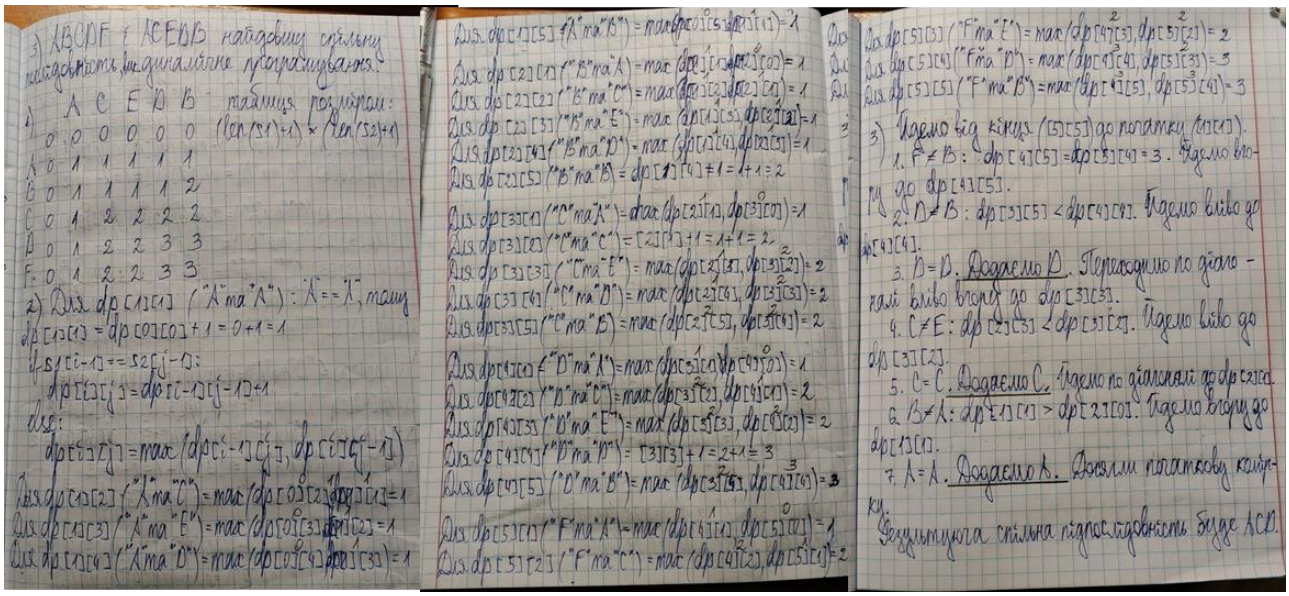

Рисунок 1 – Розв’язання

1.3 Відповіді на 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. Формально для двох рядків *X*

і Y, LCS є послідовністю Z, яка зустрічається в обох X і Y, і є найдовшою серед усіх таких послідовностей.

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

Основні методи:

- *Динамічне програмування* (табличний метод) — найпоширеніший і ефективний для знаходження LCS.
- *Рекурсія з використанням мемоізації*. Цей метод також використовує рекурсію, але зберігає проміжні результати у пам'яті (мемоізація), щоб уникнути зайвих обчислень. Він також працює з часом $O(m \cdot n)$, але може бути більш ефективним у випадках, коли є багато спільних підпослідовностей.
- *Алгоритм Хаббарда* — модифікований метод, який оптимізує класичний підхід. Його складність також $O(m \cdot n)$.
- *Алгоритм повного перебору*. У цьому методі всі можливі підпослідовності перебираються з метою знаходження найбільшої спільної підпослідовності. Проте цей підхід неефективний щодо часу виконання, оскільки має експоненціальну складність.

3. Як працює алгоритм динамічного програмування для знаходження LCS?

Алгоритм будує двовимірну таблицю dp[i][j], де:

i — індекс символу першого рядка,

j — індекс символу другого рядка.

Правила заповнення:

Якщо s1[i-1] == s2[j-1], то dp[i][j] = dp[i-1][j-1] + 1.

Інакше: dp[i][j] = max(dp[i-1][j], dp[i][j-1]).

Фінальний результат — у клітинці dp[m][n], де m і n — довжини рядків.

4. Як працює алгоритм Хаббарда для знаходження LCS?

Алгоритм Хаббарда — це вдосконалений підхід до знаходження LCS, що мінімізує використання пам'яті і зменшує кількість непотрібних порівнянь. Основні ідеї:

- Використання позицій символів у вигляді індексів.
- Побудова бітових масивів або спеціалізованих структур для швидкого знаходження збігів.
- Можна реалізувати за час O(n log n) у деяких специфічних випадках (наприклад, коли алфавіт малий).

Однак цей алгоритм складніший у реалізації і не завжди виграє на практиці при коротких рядках.

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

Критерій	**Динамічне програмування**	**Алгоритм Хаббарда**
Складність реалізації	Простий	Складний
Часова складність	O(n * m)	O(n log n) або краще в деяких випадках
Пам'ять	Вимагає таблиці розміром n × m	Оптимізована
Надійність	Універсальний метод	Підходить лише для деяких випадків

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

Задача LCS використовується в багатьох сферах:

- Порівняння текстів — знаходження схожих фрагментів, виявлення змін.
- Системи контролю версій (Git, SVN) — порівняння версій файлів.
- Біоінформатика — порівняння ДНК, білкових послідовностей.
- Пошукові системи та штучний інтелект — обробка природної мови.
- Стиснення даних — пошук повторюваних шаблонів.
- Навчальні системи — перевірка подібності відповідей студентів.