Практична робота № 7

Тема. Алгоритми на рядках

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

Постановка завдання.

Виконати індивідуальне завдання. Завдання полягає у розв'язанні задачі, яку потрібно вибрати зі списку, наведеного нижче. Номер варіанта відповідає номеру студента у списку групи. У разі, якщо було досягнуто кінця списку задач, потрібно циклічно повернутися на його початок.

Завдання №14

Для пошуку найдовшої спільної підпослідовності за допомогою алгоритму Хаббарда використовується динамічне програмування. Основна ідея полягає в тому, щоб знайти довжину найбільшої спільної підпослідовності для кожної пари префіксів двох послідовностей.

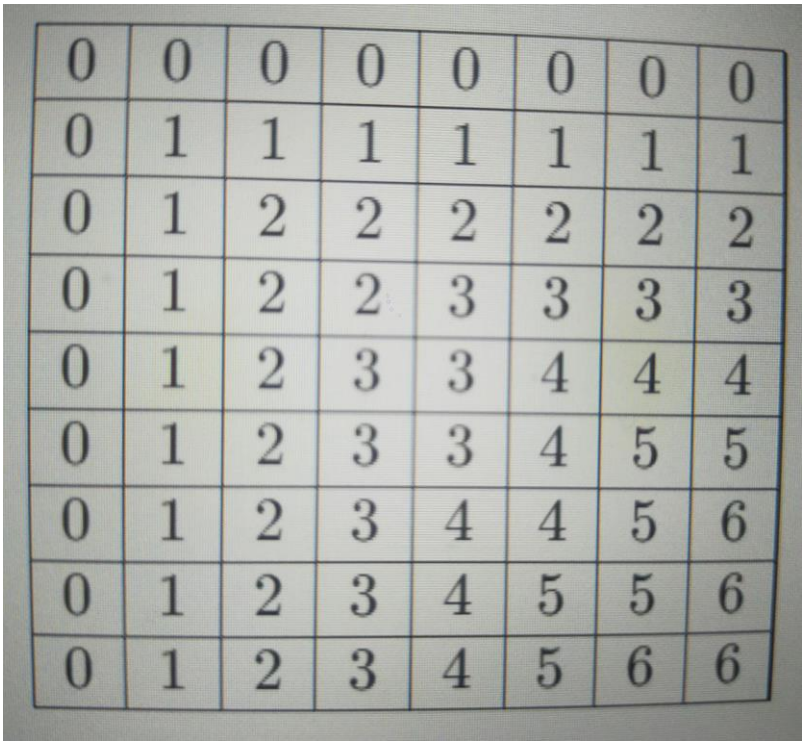
Для заданих послідовностей "ABABABAB" і "BABABABA" ми можемо створити матрицю dpdp розміром $n \times m$n×m, де nn - довжина першої послідовності, mm - довжина другої послідовності.

1. Ініціалізуємо перший рядок та першу колонку матриці dpdp нулями.
2. Проходимо по кожному елементу матриці dpdp:
    - Якщо символи на відповідних позиціях у двох послідовностях співпадають, то $dp[i][j] = dp[i-1][j-1] + 1$dp[i][j]=dp[i−1][j−1]+1.
    - Якщо символи не співпадають, то $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$dp[i][j]=max(dp[i−1][j],dp[i][j−1]).
3. Найдовша спільна підпослідовність буде рівна значенню в правому нижньому куті матриці dpdp.

Застосуємо цей алгоритм до заданих послідовностей:

Послідовність 1: "ABABABAB" Послідовність 2: "BABABABA"

4. Ініціалізуємо матрицю dpdp розміром 9×89



0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1
0	1	2	2	2	2	2	2
0	1	2	2	3	3	3	3
0	1	2	3	3	4	4	4
0	1	2	3	3	4	5	5
0	1	2	3	4	4	5	6
0	1	2	3	4	5	5	6
0	1	2	3	4	5	6	6

5. Найдовша спільна підпослідовність має довжину 6.

Отже, знайдена найдовша спільна підпослідовність: "BABABA".

Контрольні питання.

6. **Задача знаходження найдовшої спільної підпослідовності (LCS)** полягає в тому, щоб знайти найбільшу кількість символів, які співпадають у двох або більше послідовностях, і які зберігають порядок, але можуть мати пропуски.
7. Головні методи для знаходження найдовшої спільної підпослідовності включають:
    - Динамічне програмування
    - Рекурсивний алгоритм зі зрізанням (рекурсійний зберіганням проміжних результатів)
    - Алгоритми на основі матриць, такі як алгоритм Хаббарда
8. **Алгоритм динамічного програмування** для знаходження LCS працює за принципом заповнення таблиці довжиною $m×mm×n$, де mm і mn - довжини двох послідовностей. Значення в кожній клітинці таблиці визначається залежно від значень попередніх клітинок. Цей алгоритм забезпечує оптимальне рішення завдяки збереженню проміжних результатів та уникненню зайвих обчислень.
9. **Алгоритм Хаббарда** також використовує динамічне програмування, але зменшує кількість просторових ресурсів, використовуючи лише одномірний масив замість матриці. Він працює шляхом заповнення цього

масиву значеннями, які представляють довжину найдовшої спільної підпослідовності для кожної пари символів.

10. **Переваги та недоліки алгоритмів**:
    - **Динамічне програмування**:
        - Переваги: ефективний для великих послідовностей, здатний знаходити оптимальне рішення.
        - Недоліки: вимагає більше просторових ресурсів через необхідність зберігати таблицю проміжних результатів.
    - **Алгоритм Хаббарда**:
        - Переваги: ефективний з точки зору просторових ресурсів, оскільки використовує лише одномірний масив.
        - Недоліки: менш ефективний для дуже великих послідовностей через обмеження на розмір масиву.
11. **Практичні застосування задачі LCS** включають:
    - Біоінформатика: зіставлення генетичних послідовностей.
    - Редактори тексту: знаходження найбільшої спільної підпослідовності для порівняння текстових файлів.
    - Комп'ютерна графіка: визначення схожості між двома зображеннями або відеофрагментами.
    - Комп'ютерна безпека: виявлення схожих частин програмного коду для виявлення загроз.