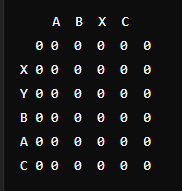
Для знаходження найдовшої спільної підпослідовності (LCS) для двох рядків "XYBAC" та "ABXC" за допомогою алгоритму динамічного програмування, виконаємо наступні кроки:

1. **Ініціалізація матриці dp**

Створимо матрицю dp розміром (len(s1) + 1) x (len(s2) + 1) і заповнимо її нулями. Рядок s1 дорівнює "XYBAC", а рядок s2 дорівнює "ABXC".

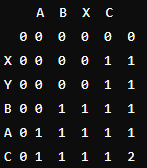


1. **Заповнення матриці dp**

Обчислимо значення для кожної комірки dp[i][j] за наступним правилом:

* Якщо символи s1[i-1] і s2[j-1] однакові, тоді dp[i][j] = dp[i-1][j-1] + 1.
* Якщо символи s1[i-1] і s2[j-1] різні, тоді dp[i][j] = max(dp[i-1][j], dp[i][j-1]).

Процес заповнення матриці:



Покрокове заповнення:

* dp[1][1] (X, A): X не дорівнює A → dp[1][1] = max(dp[0][1], dp[1][0]) = 0
* dp[1][2] (X, B): X не дорівнює B → dp[1][2] = max(dp[0][2], dp[1][1]) = 0
* dp[1][3] (X, X): X дорівнює X → dp[1][3] = dp[0][2] + 1 = 1
* dp[1][4] (X, C): X не дорівнює C → dp[1][4] = max(dp[0][4], dp[1][3]) = 1
* dp[2][1] (Y, A): Y не дорівнює A → dp[2][1] = max(dp[1][1], dp[2][0]) = 0
* dp[2][2] (Y, B): Y не дорівнює B → dp[2][2] = max(dp[1][2], dp[2][1]) = 0
* dp[2][3] (Y, X): Y не дорівнює X → dp[2][3] = max(dp[1][3], dp[2][2]) = 1
* dp[2][4] (Y, C): Y не дорівнює C → dp[2][4] = max(dp[1][4], dp[2][3]) = 1
* dp[3][1] (B, A): B не дорівнює A → dp[3][1] = max(dp[2][1], dp[3][0]) = 0
* dp[3][2] (B, B): B дорівнює B → dp[3][2] = dp[2][1] + 1 = 1
* dp[3][3] (B, X): B не дорівнює X → dp[3][3] = max(dp[2][3], dp[3][2]) = 1
* dp[3][4] (B, C): B не дорівнює C → dp[3][4] = max(dp[2][4], dp[3][3]) = 1
* dp[4][1] (A, A): A дорівнює A → dp[4][1] = dp[3][0] + 1 = 1
* dp[4][2] (A, B): A не дорівнює B → dp[4][2] = max(dp[3][2], dp[4][1]) = 1
* dp[4][3] (A, X): A не дорівнює X → dp[4][3] = max(dp[3][3], dp[4][2]) = 1
* dp[4][4] (A, C): A не дорівнює C → dp[4][4] = max(dp[3][4], dp[4][3]) = 1
* dp[5][1] (C, A): C не дорівнює A → dp[5][1] = max(dp[4][1], dp[5][0]) = 1
* dp[5][2] (C, B): C не дорівнює B → dp[5][2] = max(dp[4][2], dp[5][1]) = 1
* dp[5][3] (C, X): C не дорівнює X → dp[5][3] = max(dp[4][3], dp[5][2]) = 1
* dp[5][4] (C, C): C дорівнює C → dp[5][4] = dp[4][3] + 1 = 2

1. **Відновлення найдовшої спільної підпослідовності**

Починаючи з комірки dp[5][4], відновлюємо LCS:

* dp[5][4] = 2 (C, C): символ збігається, додаємо C до результату, переходимо до dp[4][3].
* dp[4][3] = 1 (A, X): символи не збігаються, переходимо до dp[4][2].
* dp[4][2] = 1 (A, B): символи не збігаються, переходимо до dp[3][2].
* dp[3][2] = 1 (B, B): символ збігається, додаємо B до результату, переходимо до dp[2][1].
* dp[2][1] = 0 (Y, A): переходимо до dp[1][1].

**Отримали LCS: "BC".**