

Zadaci

- Implementirati rekurzivno rešenje problema nalaženja najveće zajedničke podsekvence stringova **ALGORITAM** i **LOGARITAM** po formuli:

$$c[i, j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ c[i - 1, j - 1] + 1 & \text{if } i, j > 0 \text{ and } x_i = y_j \\ \max(c[i, j - 1], c[i - 1, j]) & \text{if } i, j > 0 \text{ and } x_i \neq y_j \end{cases}$$

Pseudokod algoritma je dat na slici 1.

```

LCS(S, n, T, m)

1  if n == 0 or m == 0
2      return 0
3  if S[n] == T[m]
4      return 1 + LCS(S, n - 1, T, m - 1)
5  else
6      return max(LCS(S, n - 1, T, m), LCS(S, n, T, m - 1))
    
```

Slika 1 - Pseudokod pronalaženja najveće zajedničke podsekvence rekurzivno

- Implementirati metodu za izračunavanje dužine najveće zajedničke podsekvence `LCS-LENGTH(X, Y)`, dinamičkim programiranjem. Parametri X i Y stringovi koji se porede. Ova metoda treba da pamti trenutnu dužinu podsekvence u svakom trenutku kao i smer iz kog je vrednost „došla“ (levo <, gore ^, levo-gore <^).
- Implementirati ispis pronađene najveće zajedničke podsekvence ulaznih stringova metodom `PRINT-LCS(b, X, i, j)`.

| | | j | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|-------|---|---|---|---|---|---|---|---|
| i | y_j | | B | D | C | A | B | A | |
| | | | | | | | | | |
| 0 | x_i | | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | A | | 0 | 0 | 0 | 0 | 1 | 1 | |
| 2 | B | | 0 | 1 | 1 | 1 | 2 | 2 | |
| 3 | C | | 0 | 1 | 1 | 2 | 2 | 2 | |
| 4 | B | | 0 | 1 | 1 | 2 | 2 | 3 | |
| 5 | D | | 0 | 1 | 2 | 2 | 2 | 3 | |
| 6 | A | | 0 | 1 | 2 | 2 | 3 | 4 | |
| 7 | B | | 0 | 1 | 2 | 2 | 3 | 4 | |

Slika 2 - Ispis najveće zajedničke podsekvence