

Trabalho: Programação Dinâmica e Complexidade (BigO)

1. Maior Subsequência Comum (*Longest Common Subsequence*)

Encontrar o comprimento da maior subsequência entre duas sequências. Uma Subsequência é uma sequência que aparece na mesma ordem, porém não necessariamente contígua.

Exemplo:

“ABCDGH” e “AEDFHR” gera “ADH” (tam = 3)

“AGGTAB” e “GXTXAYB” gera “GTAB” (tam = 4)

Abaixo segue o código recursivo ou em <https://bit.ly/3o5Ugrk> :

```
def lcs(X, Y, m, n):  
    if m == 0 or n == 0:  
        return 0;  
    elif X[m-1] == Y[n-1]:  
        return 1 + lcs(X, Y, m-1, n-1);  
    else:  
        return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));  
  
X = "AGGTAB"  
Y = "GXTXAYB"  
l = lcs(X, Y, len(X), len(Y))  
print ("Length of LCS is ",l)  
  
Length of LCS is 4
```

2. Objetivos

Desenvolva a versão dinâmica deste código.

Prove as análises BigO da implementação recursiva e dinâmica

3. Avaliação

Data	Pontuação
25/09 - às 23:59	10

- Artigo (pdf) com pseudo-códigos e análises
- **Entrega em grupo via AVA (SOMENTE)**