

1) Usando programação dinâmica, preencha uma tabela para calcular a distância de edição entre as palavras **logaritmo** e **algoritmo**.

		A	L	G	O	R	I	T	M	O	
	O	1	2	3	4	5	6	7	8	9	
L	1	1	1	2	3	4	5	6	7	8	
O	2	2	2	2	2	3	4	5	6	7	
G	3	3	3	2	3	3	4	5	6	7	
A	4	3	4	3	3	4	4	5	6	7	
R	5	4	4	4	4	3	4	5	6	7	
I	6	5	5	5	5	4	3	4	5	6	
T	7	6	6	6	6	5	4	3	4	5	
M	8	7	7	7	7	6	5	4	3	4	
O	9	8	8	8	7	7	6	5	4	3	

Dessa forma, podemos concluir que a distância de edição para transformar “logaritmo” em “algoritmo” é 3

2) Dado uma tabela preenchida com a distância de edição calculada entre duas cadeias de caracteres, $x[1..n]$ e $y[1..m]$, escreva um algoritmo que informe as operações realizadas para transformar x em y .

function imprimeOperacoes($x[1..n]$, $y[1..m]$, $E[0..n,0..m]$)

Python

```
def imprimeOperacoes(x, y, E):
    n = len(x)
    m = len(y)

    operacoes = []
    i = n-1
    j = m-1

    while i != 0 and j != 0:
        cas = E[i-1, j-1]
        ins = E[i, j-1]
        rem = E[i-1, j]

        ops = [cas, ins, rem]
        #Obtendo a operação com o menor custo
        realizado = ops.index(min(ops))

        string = ""
        if realizado == 0:
            if E[i-1, j-1] != E[i, j]:
                string = f"substituição do {x[i]} por {y[j]}"
            else:
                string = f"casamento do {x[i]} com {y[j]}"
            i = i-1
            j = j-1
        elif realizado == 1:
            string = f"inserindo {y[j]}"
            j = j-1
        else:
            string = f"removendo {x[i]}"
            i = i-1

        operacoes.append(string)

    for op in reversed(operacoes):
        print(op)

imprimeOperacoes(x, y, E)
```

Saída do código:

substituição do s por e
substituição do i por s
casamento do t com t
casamento do u com u
removendo a
casamento do d com d
casamento do o com o