

# ALGORITMO DE WAGNER FISCHER

Rodrigo E. A. Marques, Jacob A. R. L. C. Araújo

Departamento de Computação – Universidade Federal do Piauí – Teresina- Pi – Br

Disciplina: Projeto e análise de algoritmos – Universidade Federal do Piauí

[rodrigo6802@ufpi.edu.br](mailto:rodrigo6802@ufpi.edu.br), [jacobalexandre123@gmail.com](mailto:jacobalexandre123@gmail.com)

**Abstract:** This work consists of the description of Wagner Fischer algorithm, where the code will search for the shortest path between two strings (levenshtein distance).

**Resumo:** Este trabalho consiste na descrição do algoritmo de Wagner Fischer, onde o código irá fazer a busca do menor caminho entre duas strings (levenshtein distance) .

## 1.0 Descrição do algoritmo:

O algoritmo de Wagner Fischer faz parte da programação dinâmica onde consiste no calculo da menor distância de edição com base na observação de que se reservarmos uma matriz para armazenar as distâncias de edição entre todos os prefixos da primeira string e todos os prefixos da segunda string, podemos calcular os valores na matriz preenchendo a matriz e assim, encontrar a distância entre as duas strings completas como o último valor calculado.

```
word1 = 'esrdtfy'
word2 = 'aewsrfyokj'
table = []
```

**Imagem 1.** Duas strings que vão ser utilizadas para como exemplo na implementação para calcular a menor distância.

## 2.0 – Descrição da implementação:

O algoritmo recebe duas strings e é criada uma matriz que vai ser guardado as distâncias. Nessa matriz, é setado o caso base onde a última coluna e a última linha são

preenchidas com valores que vão de 0 até o tamanho das strings respectivamente. Depois disso a matriz é percorrida da direita pra esquerda e de baixo para cima calculando os valores de distância de cada célula. Esse cálculo é feito escolhendo o menor valor dentre a célula da direita, da diagonal e de baixo; e verificando se o caractere da linha e da coluna da célula são iguais. Se não for igual acrescenta-se 1 ao valor da célula escolhida sendo resultado da célula e se for não precisa acrescentar.

### **3.0 Código fonte:**

Por ser um código um pouco extenso e também para economizar espaço, foi decidido utilizar um hyperlink direto para o Google Colab, onde está o código fonte: [Código fonte](#)

### **4.0 Complexidade:**

Sendo “m” o tamanho da primeira string e “n” o tamanho da segunda string, a matriz tem tamanho  $(m+1) \times (n+1)$ . Dessa forma, como o algoritmo sempre percorre a matriz por completo a partir da penúltima linha e da penúltima coluna e faz 3 comparações para cada célula percorrida. Sendo assim, a quantidade de comparações é sempre  $3m \times 3n$ . Concluímos então que a complexidade do algoritmo é  $\theta(m \times n)$ .

Referências: WAGNER-FISCHER ALGORITHM. Wikipedia. 2022. Disponível em:

[https://en.wikipedia.org/wiki/Wagner%E2%80%93Fischer\\_algorithm](https://en.wikipedia.org/wiki/Wagner%E2%80%93Fischer_algorithm)