Distância de Edição

A distância de edição é o número mínimo de *inserções, remoções e substituições* que devem ser realizadas para fazer com que uma cadeia de caracteres A seja igual a uma segunda cadeia de caracteres B. Uma das maneiras mais comuns de resolver este problema é através do uso da **programação dinâmica**. Assim, os dados são tabelados para evitar o recálculo das várias combinações possíveis. O algoritmo para calcular a distância de edição utilizando a técnica de **programação dinâmica** é O(mn), onde m corresponde ao tamanho de A e n ao tamanho de B. O algoritmo está descrito abaixo.

1: {inicialização}
2: m = *tamanho de A*
3: n = *tamanho de B*
4: **para** $i \leftarrow 0$ **até** m **faça**
5: $\quad M(i,0) \leftarrow i$
6: **fim para**
7: **para** $j \leftarrow 0$ **até** n **faça**
8: $\quad M(0,j) \leftarrow j$
9: **fim para**
10: {calcula cada elemento levando em conta os resultados anteriores}
11: **para** $i \leftarrow 1$ **até** m **faça**
12: $\quad$ **para** $j \leftarrow 1$ **até** n **faça**
13: $\quad\quad$ **se** *A(i) = B(j)* **então**
14: $\quad\quad\quad C \leftarrow 0$
15: $\quad\quad$ **senão**
16: $\quad\quad\quad C \leftarrow 1$
17: $\quad\quad$ **fim se**
18: $\quad\quad M(i,j) \leftarrow min \begin{cases} M(i-1,j-1)+C \\ M(i-1,j)+1 \\ M(i,j-1)+1 \end{cases}$
19: $\quad$ **fim para**
20: **fim para**
21: retornar $M(m,n)$

O ponto chave do algoritmo está na linha 16, onde a recorrência foi definida. A solução do problema se da através da utilização de resultados anteriores previamente calculados e memorizados. Esta é uma características presente em todos os problemas solúveis com programação dinâmica.

Abaixo pode ser visualizada a matriz M após a execução do algoritmo

quando A é igual à "antologia" e B é igual à "antologico". Pode-se observar que o valor presente no campo inferior direito da matriz corresponde à distância de edição entre A e B.

		o	**n**	**t**	**o**	**l**	**o**	**g**	**i**	**c**	**o**
	0	1	2	3	4	5	6	7	8	9	10
a	1	**1**	2	3	4	5	6	7	8	9	10
n	2	2	**1**	2	3	4	5	6	7	8	9
t	3	3	2	**1**	2	3	4	5	6	7	8
o	4	3	3	2	**1**	2	3	4	5	6	7
l	5	4	4	3	2	**1**	2	3	4	5	6
o	6	5	5	4	3	2	**1**	2	3	4	5
g	7	6	6	5	4	3	2	**1**	2	3	4
i	8	7	7	6	5	4	3	2	**1**	**2**	3
a	9	8	8	7	6	5	4	3	2	2	**3**

A seqüência de operações que se deve realizar para igualar as duas palavras pode ser obtida através da análise dos itens em negrito na matriz. Pode-se alterar o algoritmo para armazenar em uma segunda matriz, com as mesmas dimensões de M, um indicativo de qual item foi selecionado em cada execução da linha 16. Assim, se o item selecionado foi:

- $M(i-1, j-1)$ e $M(i-1, j-1) = M(i, j)$ então não fazer nada;
- $M(i-1, j-1)$ e $M(i-1, j-1) \neq M(i, j)$ então substituir $A(i)$ por $B(j)$;
- $M(i-1, j)$ então remover $A(i)$;
- $M(i, j-1)$ então inserir $B(j)$ depois de $A(i)$.