编辑距离问题

问题描述：

给定两个单词 *word1* 和 *word2*，计算出将 *word1* 转换成 *word2* 所使用的最少操作数 。

你可以对一个单词进行如下三种操作：

1. 插入一个字符
2. 删除一个字符
3. 替换一个字符

问题分析：

现在我们要找到从字符串 A 转换成 B 的子结构，假设字符序列 $A_1A_2\cdots A_i$ 和 $B_1B_2\cdots B_j$ 是 A, B 的前 i, j 个字符构成的子串，下面考虑怎么由他的前驱状态进行上述的三种操作达到当前状态，即他的状态转移方程。

1．插入操作

考虑字符序列 $A_1A_2\cdots A_i$ 和 $B_1B_2\cdots B_{j-1}$，设此时的操作数为 op_1，那么要匹配字符 B_j，我们可以在 A_i 和 A_{i+1} 之前插入字符 $A_i'=B_j$

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_i & A_i' \\ B_1 & B_2 & \cdots & B_{j\text{-}1} & B_j \end{bmatrix}$$

得到状态转移方程为 $op = op_1 + 1$

2．删除操作

考虑字符序列 $A_1A_2\cdots A_{i\text{-}1}$ 和 $B_1B_2\cdots B_j$，设此时的操作数为 op_2，那么要匹配字符 A_i，我们可以删掉 A_i

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_{i\text{-}1} & \times \\ B_1 & B_2 & \cdots & B_{j\text{-}1} & B_j \end{bmatrix}$$

得到状态转移方程为 $op = op_2 + 1$

2．删除操作

考虑字符序列 $A_1A_2\cdots A_{i\text{-}1}$ 和 $B_1B_2\cdots B_{j\text{-}1}$，设此时的操作数为 op_3，那么要匹配字符 A_i 和 B_j ，如果他们不等的话我们可以修改 A_i 为 $A_i'=B_j$

$$\begin{bmatrix} A_1\, A_2\ \cdots\ A_{i\text{-}1}\, A_i{}' \\ B_1\, B_2\ \cdots\ B_{j\text{-}1}\, B_j \end{bmatrix}$$

得到状态转移方程为$op = op_3 + 1$

现在给出编辑问题的状态转移方程：

定义 $lev_{A,B}(i, j)$ 为字符串 A 在 i 位置和字符串 B 在 j 位置时进行转换的操作数，我们可以得到

$$lev_{A,B}(i,j) = \begin{cases} \max(i,j), & if\,(\min(i,j)=0) \\ \min\begin{cases} lev_{A,B}(i-1,j)+1 \\ lev_{A,B}(i,j-1)+1 \\ lev_{A,B}(i-1,j-1)+1, \quad if\,(A_i \,!\!= B_j) \end{cases} & otherwise \end{cases}$$