

第四次作业

作业1：输入为**CRONALDO**和**RENATO**，填写计算二者最小编辑距离的矩阵**E**和**B**

	y_j	R	E	N	A	T	O
x_i	0	1	2	3	4	5	6
C	1	↖1	↖2	↖3	↖4	↖5	↖6
R	2	↖1	↖2	↖3	↖4	↖5	↖6
O	3	↑2	↖2	↖3	↖4	↖5	↖5
N	4	↑3	↖3	↖2	←3	←4	←5
A	5	↑4	↖4	↑3	↖2	←3	←4
L	6	↑5	↖5	↑4	↑3	↖3	↖4
D	7	↑6	↖6	↑5	↑4	↖4	↖4
O	8	↑7	↖7	↑6	↑5	↖5	↖4

作业2：完成最小编辑距离的优化解的构造算法，输入**X**和**Y**，输出基本操作序列，将**X**转换为**Y**

基本操作接口：

```
1 (1) Insert(pos, a)
2 (2) Modify(pos, b)
3 (3) Delete(pos)
```

定义优化构造算法输出序列的算法：

- 打印构造信息

```
1 printConstructInfo(x,y){
2     (E,B) ← MinimalEditDistance(x,y); //获取最优解的信息
3     m ← len(x);
4     n ← len(y);
5     result ← E[m,n];
6     print(result);
7     printDetailInfo(x,y,B);
8 }
```

- 打印详细信息

```
1 printDetailInfo(x,y,B){
2     m ← len(x);
3     n ← len(y);
4     if B[m,n]='<'{
5         print(x[1:m-1],y[1:n-1],B); //先打印前面的构造信息
6         if x[m-1]≠y[n-1]{
7             print(Modify(m,y[n])); //如果两者不同,说明需修改。打印信息: x的第m个位置被修改为了y[n]
8         }
9     }
10    else if B[m,n]='<←'{
11        print(x[1:m],y[1:n-1],B);
12        print(Insert(m,y[n])); //打印信息: x的第m个位置插入了y[n]
13    }
14    else{
15        print(x[1:m-1],y[1:n],B);
16        print>Delete(m)); //打印信息: x的第m个位置被删除了
17    }
18 }
```

时间复杂度分析:

m是x数组长度, **n**是y数组长度。

1) 时间复杂性:

`MinimalEditDistance(x,y)` 的时间复杂性为 $O(mn)$, 构造最优解的时间复杂性为 $O(m+n)$ (最差情况走一个直角, 比如先左再上); 总时间复杂性为 $O(mn)$ 。

2) 空间复杂性:

需要数组E和B; E是 $(n+1)(m+1)$ 的, B是m n的, 所以是 $O(mn)$

和之前一样B可以省略（在 $O(1)$ ）时间内可以判断出方向，所以可以忽略B这个数组。同时，因为计算的时候只用到两行E的信息，所以可以只存两行或者两列的E信息。总可以优化为 $O(\min\{m,n\})$ 的。