

CLRS Problem 15-5(Edit Distance) Report

洪方舟

2016013259

Email: hongfz16@163.com

2018 年 3 月 24 日

1. 实验目的

- 实现计算编辑距离的算法，重构出编辑过程，并且给出方便测试的用户界面
- 给出编辑距离算法与最优对齐问题的等价性

2. 实验环境

操作系统: Windows 10

处理器: Intel Core i7-7700k CPU @ 4.20GHz × 8

编程语言: C++

开发环境: Visual Studio

3. 算法说明

本问题是一个动态规划问题，定义 $COST[i][j]$ 为 $SourceString.substring(0, i)$ 转化为 $TargetString.substring(0, j)$ 的编辑距离，那么最终要求的就是 $COST[m][n]$ ，下面分几种情况讨论状态转移：

- $SourceString[i] = TargetString[j]$, $COST[i][j] = COST[i-1][j-1] + COSTOFCOPY$
- $SourceString[i] \neq TargetString[j]$, $COST[i][j] = \min\{COST[i][j], COST[i-1][j] + COSTOFDELETE, COST[i][j-1] + COSTOFINSERT, COST[i-1][j-1] + COSTOFREPLACE\}$
- $SourceString[i] = TargetString[j-1] \&\& SourceString[i-1] = TargetString[j]$, $COST[i][j] = \min\{COST[i][j], COST[i-2][j-2] + COSTOFTWIDDLE\}$

最后需要找出 $COST[i][n] + COSTOFKILL$ 的最小值以确定在何处 $KILL$

4. 等价性说明

实际上编辑距离问题是最优对齐问题更为一般的问题，只需要令

$$COSTOFCOPY = -1$$

$$COSTOFREPLACE = 1$$

$$COSTOFINSERT = 2$$

$$COSTOFDELETE = 2$$

不允许 $TWIDDLE$ 和 $KILL$ 操作，则编辑距离问题就转化为最优对齐问题

5. 实验方法

编写计算编辑距离的算法接口，使用 *QT* 编写图形界面，调用算法接口；

6. 实验结果

得到一个方便测试的计算编辑距离算法的图形界面，操作方式如下：在 *SourceString* 和 *TargetString* 两栏中分别填入需要计算编辑距离的两个字符串，在接下去的六个框中填入用户需要的各操作的代价，点击 *Calculate* 按钮，将会给出编辑距离，并且给出每一步的具体操作以及操作结果。

7. 源代码与可执行文件说明

源代码存放在 *src/EditDistQt/* 文件夹下，可执行文件存放在 *bin/EditDistGUI*