自动文字校对动态规划算法的设计与实现

The Design and Application of A Dynamic Program Algorithm in Automatic Text Collating

徐连诚　石　磊

（山东师范大学计算机科学系　济南250014）

Abstract　Automatic text collating is an important problem of the test in Computer Knowledge and Skills. In this paper, we discuss elements of dynamic program algorithm, then, we settle out a sample of automatic text collating with this algorithm.

Keywords　Automatic text collating, Dynamic program algorithm, Test in computer knowledge and skill, Automatic judgment

1. 引言

近年来，计算机知识与技能考试逐渐向自动化方向发展，其中所涉及的核心技术之一是试题自动阅卷。

计算机知识与技能考试的试题类型主要包括客观题（如选择、判断、填空、识别等）和主观题（如文字录入、Windows操作、Word文档编辑、Excel表格处理、PowerPoint幻灯片制作、Internet应用、网页设计等）两大部分；从阅卷评判来看，主观题的自动阅卷相对复杂，本文主要就文字录入的自动阅卷进行讨论。

文字录入考察考生对一段文字在一定时间内输入文字的准确率，自动文字校对实际上是对考生输入的有序字符集与标准答案的有序字符集进行比对和匹配，求出其最大公共有序字符集，根据该字符集和标准答案字符集的比率确定试题的得分率。如何求解这个最大公共有序字符集，是问题的关键。

2. 动态规划算法的基本思路

2.1　动态规划算法简介

动态规划算法是将待求解问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解[1]。适合于动态规划法求解的问题，经分解得到的子问题往往不是互相独立的，这是与分而治之的方法——分治法所不同的。若采用分治法来解决这类问题，由于分解出的子问题数目太多，以致于最后解决原问题需要耗费指数时间。而实际不同子问题的数目只有多项式数量级[2]，这主要是因为有些子问题被重复计算了多次，如果在搜索子问题的过程中能够保存已经计算的答案（即填写表格）并在需要的时候直接找出此答案，就可以避免大量的重复计算，这就是动态规划的基本思想。

2.2　动态规划算法的基本要素

从一般意义上讲，对于可分解的问题如果使用动态规划算法，问题本身具有最优子结构性质和子问题重叠性质。也就是说，这两个性质是该类问题能用动态规划算法求解的基本要素。

1）*最优子结构性质*　当问题的最优解包含了其子问题的最优解时，称该问题具有最优子结构性质。问题的最优子结构性质为问题的动态规划求解提供了重要线索。在动态规划中，这个性质使我们能够以自底向上的方式递归地从子问题的最优解逐步构造出整个问题的解，同时它也使我们能够在相对小的子空间中考虑问题。

2）*子问题的重叠性质*　在使用动态规划算法递归地自顶向下求解问题时，每次产生的子问题并不总是没有进行过求解的新问题，有些是被反复计算多次的（如图1所示）。

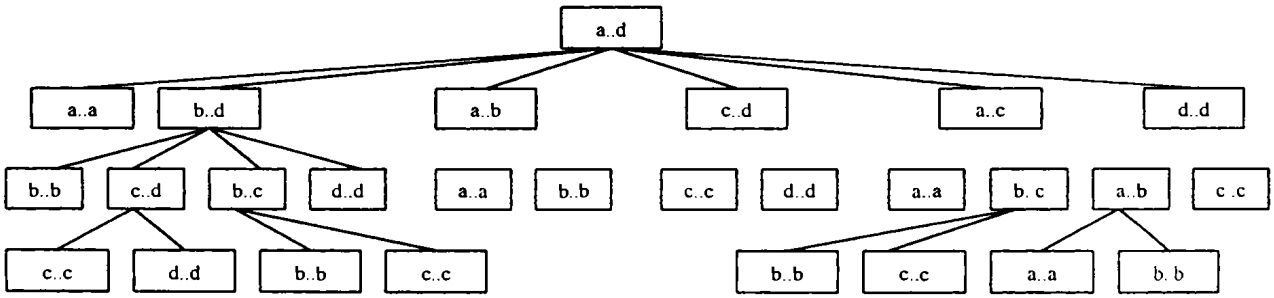

图1　计算a..d的递归树

因此在实际的计算中，对每个子问题只求解一次并把它保存在一个表格中，当再次需要这个子问题的解时，只要简单查出结果就可以了，而这仅需要多项式的计算时间。

2.3　动态规划算法的基本过程

设计一个动态规划算法可以按照以下步骤进行：

1）找出问题最优解的性质，刻画出其结构特征，并递归地定义出最优值；

2）以自底向上的方式计算出最优值；

3）根据最优值的求解信息构造最优解。

3. 算法的设计与实现

3.1　问题特征

最大公共有序字符集问题可以描述成以下的一般形式：

1)一个给定有序字符集的子集是在该字符集删去若干元素后得到的子集，即若给定字符集 $L=\{l_1,l_2,\cdots,l_n\}$，则另一字符集 $Z=\{z_1,z_2,\cdots,z_k\}$ 是 L 的子集是指存在一个严格递增下标序列 $\{i_1,i_2,\cdots,i_k\}$ 使得对于所有的 $j=1,2,\cdots,k$ 有 $z_j=x_{i_j}$；

2)给定两个有序字符集 A 和 B，当另一个字符集 Z 既是 A 的子集又是 B 的子集时，称 Z 是有序字符集 A 和 B 的公共子集；

3)最大公共有序字符集问题是指：给定两个有序字符集 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$，找出 A 和 B 的一个最大公共子集。

最大公共有序字符集问题的最容易想到的求解方法是穷举搜索法，即对 A 的所有子集，检查它是否同时也是 B 的子集，从而确定它是否为 A 和 B 的公共子集，若是则同时记录最长的公共子集，这样对 A 的所有子集都检查过之后即可求出 A 和 B 的最大公共有序字符集。对于含有 m 个元素的字符集 A 来说，其所有可能的子集个数为 2^m，从而穷举搜索法的计算时间是指数时间[3]，显然在实际使用中这是很不理想的。

3.2 问题结构分析

事实上，设有序字符集 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$ 的一个最大公共有序字符集为 $Z=\{z_1,z_2,\cdots,z_k\}$，则最大公共有序字符集问题具有下面的性质：

1)若 $a_m=b_n$，则 $z_k=a_m=b_n$，且 Z_{k-1} 是 A_{m-1} 和 B_{n-1} 的最大公共有序字符集

2)若 $a_m\neq b_n$，且 $z_k\neq a_m$，则 Z 是 A_{m-1} 和 B 的最大公共有序字符集

3)若 $a_m\neq b_n$，且 $z_k\neq b_n$，则 Z 是 A 和 B_{n-1} 的最大公共有序字符集

其中，$A_{m-1}=\{a_1,a_2,\cdots,a_{m-1}\}$，$B_{n-1}=\{b_1,b_2,\cdots,b_{n-1}\}$，$Z_{k-1}=\{z_1,z_2,\cdots,z_{k-1}\}$

最大公共有序字符集问题的这种性质即为最优子结构性质。

由此可知，要求解 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$ 的最大公共有序字符集，可以按照以下递归方式进行：

1)当 $a_m=b_n$ 时，找出 A_{m-1} 和 B_{n-1} 的最大公共有序字符集，然后在其尾部加上 $a_m(=b_n)$ 即可得出 A 和 B 的最大公共有序字符集；

2)当 $a_m\neq b_n$ 时，则 A_{m-1} 和 B_n 的最大公共有序字符集与 A_m 和 B_{n-1} 的最大公共有序字符集中较大者即为 A 和 B 的最大公共有序字符集。

若设 $C_{i,j}$ 为 A_i 和 B_j 的最大公共有序字符集的宽度(即元素个数)，则有

$$C_{i,j}=\begin{cases}0 & i=0\text{或 }j=0\\ C_{i-1,j-1}+1 & i,j>0\text{且 }a_i=b_j\\ \max\{C_{i,j-1},C_{i-1,j}\} & i,j>0\text{且 }a_i\neq b_j\end{cases}$$

3.3 计算最优值

```
Const TextLenM = 255 '有序字符集的元素最大个数
Private a(TextLenM) As String * 1' 有序字符集 A
Private b(TextLenM) As String * 1' 有序字符集 B
Private C(TextLenM, TextLenM) As Integer
  '有序字符集 Ai 和 Bj 的最大公共有序字符集元素个数
Private d(TextLenM, TextLenM) As Integer ' 最优解标记
Private Sub TextOV(m As Integer, n As Integer)' 求解最优值
  Dim i As Integer, j As Integer
  For i=1 To m
    C(i,0)=0
  Next i
  For i = 1 To n
    C(0,i)=0
  Next i
  For i = 1 To m
    For j = 1 To n
      If a(i)=b(j) Then
        C(i,j)=C(i-1, j-1)+ 1
        d(i,j)=1
      ElseIf C(i-1,j)>= C(i,j-1) Then
        C(i,j)=C(i-1,j)
        d(i,j)=2
      Else
        C(i,j)=C(i,j-1)
        d(i,j)=3
      End If
    Next j
  Next i
End Sub
```

3.4 构造最优解

```
Private k As Integer 'Initial Value is 1
Private e(TextLenM) As String * 1' 有序字符集 Z
Private TextOR(i As Integer, j As Integer)
  If i = 0 Or j=0 Then Exit Sub
  If d(i, j)=1 Then
    TextOR(i-1, j-1)
    e(k)=a(i)
    k=k+1
  ElseIf d(i,j)=2 Then
    TextOR(i-1,j)
  Else
    TextOR(i,j-1)
  End If
End Sub
```

4. 算法效率及改进

4.1 算法效率

计算最优值的算法时间复杂性为 $O(mn)$，空间复杂性为 $O(mn)$。

构造最优解的算法时间复杂性为 $O(m+n)$，空间复杂性为 $Max\{O(m),O(n)\}$。

4.2 算法改进

对于一个具体问题，按照一般的算法设计策略设计出来的算法，往往在算法的时间复杂性上还有较大的改进余地，通常可以利用具体问题的一些特殊性对算法作进一步的改进。

例如，在计算最优值算法和构造最优解算法中，可以进一步省略数组 d。事实上，数组元素 $C(i,j)$ 的值仅由 $C(i-1,j-1)$，$C(i-1,j)$ 和 $C(i,j-1)$ 这三个数组元素的值所确定。对于给定的数组元素 $C(i,j)$，我们可以不借助于数组 d 而仅借助于 C 本身在 $O(1)$ 时间内确定 $C(i,j)$ 的值是由 $C(i-1,j-1)$，$C(i-1,j)$ 和 $C(i,j-1)$ 中哪一个值所确定的。从而可以节省 $\theta(mn)$ 的空间。虽然数组 C 仍需要 $\theta(mn)$ 的空间使得在渐近意义上算法仍需要 $\theta(mn)$ 的空间，但上述方法对空间复杂性的常数因子进行了改进。

结束语 计算机知识与技能考试是一个古老而又崭新的课题，涉及到的领域广泛而深远，有待优化的方面还非常繁杂。本文仅就其中文字录入的自动阅卷评判算法进行了一定的探讨，实际上，Windows 操作、Word 处理等试题的相关语法描述和自动阅卷也是一类非常有意义的研究课题。

参 考 文 献

1 王晓东. 计算计算法设计与分析. 北京：电子工业出版社，2001.1

2 Edelsbrunner H . Algorithms in Combinatorial Geometry. Volume 10 of EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1987

3 张莹. 运筹学基础. 北京：清华大学出版社，1995