Assignment 2 Supplement
Algorithm Design and Analysis

bitjoy.net

January 21, 2016

3 Partition

设 $dp[i][j]$ 表示 $s[i,...,j]$ 是否为回文串，$cut[j]$ 表示 $s[1,...,j]$ 需要多少刀才能切成回文串，则我们需要求 $cut[n]$。

对于 $cut[j]$，切第一刀的时候，我们尝试前面 j 个切割位点，比如我们尝试切第 i 点，则 $s[1,...,j]$ 被分成了 $s[1,...,i-1]$ 和 $s[i,...,j]$，如果 $s[i,...,j]$ 为回文串，则 $s[i,...,j]$ 不需要再切了，又因为 $cut[i-1]$ 我们之前已经算过了，所以在 i 处切一刀，有 $cut[j] = cut[i-1] + 1$。尝试所有的切割位点 i，找最小的 $cut[j]$。

如果发现 $i = 1$ 的时候 $s[i,...,j]$ 为回文串，则 $s[1,...,j]$ 不需要再切了，所以 $cut[j] = 0$。

DP 转移公式如下：

$$cut[j] = \min_{i \in \{i \mid s[i,...,j] \ is \ palindromic\}} \{cut[j], cut[i-1] + 1\}$$

因为当 $i == j$ 时，只有一个字母，$s[j]$ 一定为回文，所以至少有一个 i 可满足上式。

```
PARTITION(s)
 1  n = length(s)
 2  dp[n][n] = false
 3  for i = 1 to n
 4        dp[i][i] = true
 5  for j = 1 to n
 6        cut[j] = j // set maximum # of cut
 7        for i = 1 to j
 8              if (s[i] == s[j]) && (j - i <= 1 || dp[i + 1][j - 1])
 9                    dp[i][j] = true
10                    if i > 1
11                          cut[j] = min(cut[j],cut[i - 1] + 1)
12                    elseif i == 1
13                          cut[j] = 0
14  return cut[n]
```

时间复杂度为 $O(n^2)$。

4 Subsequence Counting

设 $dp[i][j]$ 表示由 $S[1,...,i]$ 变到 $T[1,...,j]$ 有多少种 subsequence 方法。此时我们有两种选择，如果 $S[i] \neq T[j]$，则等价于由 $S[1,...,i-1]$ 变到 $T[1,...,j]$，所以 $dp[i][j] = dp[i-1][j]$；如果 $S[i] = T[j]$，则还可以由 $S[1,...,i-1]$ 变到 $T[1,...,j-1]$，所以 $dp[i][j] += dp[i-1][j-1]$。

DP 转移公式如下：

$$dp[i][j] = \begin{cases} dp[i-1][j] & \text{if } S[i] \neq T[j] \\ dp[i-1][j] + dp[i-1][j-1] & \text{if } S[i] = T[j] \end{cases}$$

初始值 $dp[i][0] = 1$，因为由任意字符串转换为空字符串只有一种方法，就是把 S 中的字符全删了；$dp[0][j] = 0$，因为空字符串的 subsequence 不可能是是某个非空字符串。

SUBSEQ-COUNTING(S, T)

```
1  n = length(S); m = length(T)
2  dp[n][m] = 0
3  for i = 0 to n
4       dp[i][0]=1
5  for i = 1 to n
6       for j = 1 to m
7            dp[i][j] = dp[i-1][j]
8            if S[i] == T[j]
9                 dp[i][j] += dp[i-1][j-1]
10 return dp[n][m]
```

时间复杂度为 $O(nm)$。