算法设计与分析 LECTURE 8

第九周作业-solution

LECTURER: 杨启哲

LAST MODIFIED: 2024 年 1 月 2 日

1. (教材习题 6.5) 请说明如何修改我们课上提到的最长公共子序列的算法，使得其还能输出对应的最长公共子序列。

解答. 首先回顾一下最长公共子序列的算法：

最长公共子序列

输入: 两个字符串 A, B，长度分别为 m 和 n
输出: A 和 B 的最长公共子序列的长度

```
for i = 1 to m do
    L[i][0] ← 0
end for
for j = 1 to n do
    L[0][j] ← 0
end for
for i = 1 to m do
    for j = 1 to n do
        if A[i] = B[j] then
            L[i][j] ← L[i − 1][j − 1] + 1
        else
            L[i][j] ← max{L[i − 1][j], L[i][j − 1]}
        end if
    end for
end for
return L[m][n]
```

因此我们可以直接根据这个反向构建出最长公共子序列，具体如下：

(1) 从 $L[m][n]$ 开始，如果 $A[m] = B[n]$，则将 $A[m]$ 加入到最长公共子序列中，然后将指针向左上方移动一格，即 $L[m-1][n-1]$。
(2) 如果 $A[m] \neq B[n]$，则比较 $L[m-1][n]$ 和 $L[m][n-1]$，如果前者大，则将指针向上移动一格，即 $L[m-1][n]$，否则向左移动一格，即 $L[m][n-1]$。
(3) 重复上述过程，直到指针指向 $L[0][0]$。
(4) 将得到的序列逆序输出即为最长公共子序列。

□

2. (最大相连子序列) 列表 S 的相连子序列是由 S 中相邻元素构成的子序列。举例来说，若 S 为:

$$5, 15, -30, 10, -5, 40, 10$$

则 $15. - 30.10$ 是一个相连子序列，而 $5, 15, 40$ 不是。请给出针对如下任务的线性时间算法：

输入： 数值列表：$a_1, \ldots a_n$

输出： 和最大的相连子序列（假设长度为 0 的子序列的和为 0）

解答. 这里其实只要注意到，如果一个子序列的和为负数，那么这个子序列一定不会是最大的相连子序列的前缀。即令 $Sum[i]$ 为以 i 结尾的最大相连子序列的值，则 $Sum[i]$ 具有如下关系：

$$Sum[i] = \max\{a_i, Sum[i-1] + a_i\}$$

记录下当前最大和的头尾即可记录当前的子序列。进一步我们可以优化空间，只用常数个变量保存相应的值，具体如下：

最大相连子序列

输入: 数值列表 $a_1, \ldots, a_n$
输出: 和最大的相连子序列

```
start ← 1, tmp_start ← 1, end ← 1
sum ← 0
max ← 0
for i = 1 to n do
    sum ← sum + a_i
    if sum < 0 then
        sum ← 0
        tmp_start ← i + 1
    end if
    if sum > max then
        max ← sum
        start ← tmp_start
        end ← i
    end if
end for
if max = 0 then
    return NULL
end if
return a_start, ..., a_end
```

该算法的时间复杂度为 $O(n)$，因为只有一层循环次数为 n 的循环，同时每次循环的操作次数是常数次。 □

3. （长途旅行）假设您准备开始一次长途旅行。以 0 英里为起点，一路上一共有 N 座旅店，距离起点的英里数分别为 $a_1 < a_2 < \ldots < a_n$。旅途中，您只能在这些旅店中停留，当然在哪停留完全由您决定。最后一座旅店 a_N 是您的终点。

   理想情况下，您每天可以行进 200 英里，不过考虑到旅店的实际距离，有时候可能还达不到这么远。假设某天您走了 x 英里，那么您将受到 $(200 - x)^2$ 的惩罚。您需要计划好行程，使得总的惩罚-每天所受惩罚的总和最小。请给出一个高效的算法，来确定一路上最优的停留位置。

解答. 记 $L[k]$ 表示到达第 k 座旅店所受到的最小惩罚，初始我们令 $L[0]=0$，即为起点。则我们可以得到如下的关系式：

$$L[k]=\min_{j<k,a_k-a_j\le 200} L[j]+(200-a_k+a_j)^2$$

根据上述公式更新 $L[k]$ 即可获取最终答案。注意到 200 是一个常数，从而每次循环执行的比较次数也是常数 (只跟 200 有关)，所以算法的时间复杂性为 $O(n)$. □