第 3-1 讲: 动态规划

姓名: 赵超懿　　**学号:** 191870271

评分: ______　　**评阅:** ______

2020 年 9 月 12 日

请独立完成作业，不得抄袭。
若得到他人帮助, 请致谢。
若参考了其它资料，请给出引用。
鼓励讨论，但需独立书写解题过程。

1 作业（必做部分）

题目 1 (TC 15.1-1)
Show that equation (15.4) follows from equation (15.3) and the initial condition T(0)=1.

解答:
记 $\sum\limits_{i=0}^{n} = S(n)$,
$T(n) = 1 + \sum\limits_{i=0}^{n-1} T(i) \Leftrightarrow S(n) - S(n-1) = 1 + S(n-1)$,
解得 $S(n) = 2^{n+1} - 1 \Rightarrow T(n) = 2^n$.

题目 2 (TC 15.1-3)
Consider a modification of the rod-cutting problem in which, in addition to a price p_i for each rod, each cut incurs a fixed cost of c. The revenue associated with a solution is now the sum of the prices of the pieces minus the costs of making the cuts. Give a dynamic-programming algorithm to solve this modified problem.

解答:
```
CUT-ROD(p,n)
    r[0]=0;
    for i=1 to n
        q=p[j]
        for j=1 to i-1
            q=max(q,p[j]+r[i-j]-c)
        r[i]=q
    return r[n]
```

题目 3 (TC 15.2-2)
Give a recursive algorithm MATRIX-CHAIN-MULTIPLY(A,s,i,j) that actually performs the optimal matrix-chain multiplication, given the sequence of matrices $< A1, A2, ..., An_i >$,the stable computed by MAT R I X-CHAIN-ORDER, and the indices i and j. (The initial call would be MAT R I X-CHAIN-MULTIPLY(A,s,1,n).

解答:
MATRIX-CHAIN-MULTIPLY(A,s,i,j)
 if i == j
  return A_i
 if i == j-1
  return $A_i \times A_j$
 return MATRIX-CHAIN-MULTIPLY(A,s,i,s[i,j])× MATRIX-CHAIN-MULTIPLY(A,s,s[i,j]+1,j)

题目 4 (TC 15.2-4)
Describe the subproblem graph for matrix-chain multiplication with an input chain of length n. How many vertices does it have? How many edges does it have, and which edges are they?

解答:
包含 $C_n^2 + n = \dfrac{n(n+1)}{2}$ 个顶点
包含 $\sum\limits_{i=1}^{n}\sum\limits_{i\leqslant j}^{n} 2(j-i) = \dfrac{n(n-1)(n+1)}{3}$, 对于每个顶点 (i,j)$(i \leqslant j)$, 都和 $(i, k_1)(i \leqslant k_1 < j)$ 或 $(k_2, j)(i < k_2 \leqslant j)$ 连接。

题目 5 (TC 15.3-3)
Consider a variant of the matrix-chain multiplication problem in which the goal is to parenthesize the sequence of matrices so as to maximize, rather than minimize,the number of scalar multiplications. Does this problem exhibit optimal substructure?

解答:
具有最优子结构，对于每个划分，计算的次数及划分成的两部分的最大值之和加上最后乘起来的代价。最大计算量必然包括子问题的最大计算量。反证法即可证明：若最优解划分的子问题中有一个非最优解，则子问题使用最优解会得到更大的值，矛盾，故最优解包括子问题的最优解。

题目 6 (TC 15.3-5)
Suppose that in the rod-cutting problem of Section 15.1, we also had limit l_i on the number of pieces of length i that we are allowed to produce, for i=1,2...,n.Show that the optimal-substructure property described in Section 15.1 no longer holds.

解答:
反证法：假设仍然满足最优子结构。
则一次划分后得到的两个子问题仍然是最优解，当两侧最优解中长度为 k 的数量超过 l_k 时，则必然有一侧不具有最优解，矛盾。故不满足最优解。

题目 7 (TC 15.3-6)
Imagine that you wish to exchange one currency for another. You realize that instead of directly exchanging one currency for another, you might be better off making a series of trades through other currencies, winding up with the currency you want. Suppose that you can trade n different currencies, numbered 1;2;...;n,where you start with currency 1 and wish to wind up with currency n.You are given, for each pair of currencies i and j, an exchange rate r_{ij}, meaning that if you start with d units of currency i, you can trade for dr_{ij}units of currency j.A sequence of trades may entail a commission, which depends on the number of trades you make. Let c_k be the commission that you are charged when you make k trades. Show that, if c_k=0 for all k=1;2;...;n, then the problem of finding the best sequence of exchanges from currency 1 to currency n exhibits optimal sub-structure. Then show that if commissions c_k are arbitrary values, then the problem of finding the best sequence of exchanges from currency 1 to currency n does not necessarily exhibit optimal substructure.

解答:
一.
当 $\forall k \in N, c_k = 0$ 时，将问题变为从一号货币通过一次兑换兑换为 m 号货币再从 m 号货币通若干次次兑换为 n 号货币，当问题具有最优解时，假设第一次兑换为 m 号货币，从 m 号货币兑换为 n 号货币必然也是最优解。最优解结构为 solve(i)，指从 i 号货币通过 j 次兑换兑换为 n 号货币。,$solve(1) = max_{i\neq 1}(solve(i) + dr_{1,i})$
二.
当 c_k 为任意值时, 例如共有 3 中货币 1,2,3, $r_{1,2} = 2, r_{2,3} = 3, r_{1,3} = 5$,$c_1 = 1, c_2 = 10$, 当初始为 1 号货币为 d=2 时，直接兑换为三号货币结果为 9，兑换为 2 号货币再兑换为 3 号货币为 2，所以此时不具有最优子结构，原因是随着交易次数增加费用增加。

题目 8 (TC 15.4-3)
Give a memoized version of LCS-LENGTH that runs in O(mn) time.

解答:
```
m=X.length
n=Y.length
c[i,j]=-1;
LCS-LENGTH(m,n)
    if m=0||n=0
        return 0;
    if c[m,n] ⩾ 0
        return c[m,n];
    if x_m = y_n
        c[m,n]=LCS-LENGTH(m-1,n-1)+1
        b[m,n]=↖
    else
        p=LCS-LENGTH(m,n-1); q=LCS-LENGTH(m-1,n)
        c[m,n]=max(p,q)
        b[m,n]=p>q?←:↑
    return c[m,n];
```

题目 9 (TC 15.4-5)
Give an $O(n^2)$-time algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers.

解答:

A: 输入的序列;
s[i]: 以 A[i] 结尾的 A 中前 i 项的最大单调递增子序列。
初始化都为 0

```
subsequence(x)
    for i=2 to n
        for j=1 to i-1
            if A[i] > A[j]
                s[i]=max(s[i],s[j]+1)
output
    找到 s 中的最大值 s[k]，记 temp=s[k]
    while(temp > 0)
        b[s[k]]=A[k]
        temp=temp-1;
        while(k > 1)
            k–;
            if s[k]==temp && A[k] < b[s[k+1]]
                break;
```

b 数组中的 1-s[k] 的值即为最大单调子序列。

题目 10 (TC 15.5-1)

Write pseudocode for the procedure CONSTRUCT-OPTIMAL-BST(root)which,given the table root, outputs the structure of an optimal binary search tree. For the example in Figure 15.10, your procedure should print out the structure corresponding to the optimal binary search tree shown in Figure 15.9(b).

解答:

output $k_{root[1,n]}$ is the root
construct(i,j)
    if i==j
        output $d_{i-1}$is the left son of k_i, $d_i$is the right son of k_i;return
    if root[i,j]==i
        output d_{i-1} is the left son of k_i;return
    if root[i,j]==j
        output d_j is the right son of k_j;return
    output $d_{root[i,root[i,j]-1]}$ is the left son of $k_{root[i,j]}$
    output $d_{root[root[i,j]+1],j]}$ is the right son of $k_{root[i,j]}$
    construct(i,root[i,j]-1);
    construct(root[i,j]+1,j);

---

2 作业（选做部分）

题目 1 (TC Problem 15-4: Printing neatly)

解答:

---

3 Open Topics

Open Topics 1 (通信系统)
某个通信系统由 n 个设备串联构成，每个设备可能有多个厂商生产，均有带宽和价格参数。系统的总带宽决定于某个设备的最小带宽，总价格是各个设备的价格总和。请你设计一个算法，以“带宽/造价”为最优目标，确定该通信系统的构成

请按照“最优子结构确定、确定递归表达式、非递归实现”步骤完成设计和讲解.

Open Topics 2 (TC Problem 15-3: Bitonic euclidean traveling-salesman problem)

4 反馈