Dynamic Programming

2-1 Rod-cutting Problem: Given a rod of total length N inches and a table of selling prices P_L for lengths $L = 1, 2, \cdots, M$. You are asked to find the maximum revenue R_N obtainable by cutting up the rod and selling the pieces. For example, based on the following table of prices, if we are to sell an 8-inch rod, the optimal solution is to cut it into two pieces of lengths 2 and 6, which produces revenue $R_8 = P_2 + P_6 = 5 + 17 = 22$. And if we are to sell a 3-inch rod, the best way is not to cut it at all.

Length L	1	2	3	4	5	6	7	8	9	10
Price P_L	1	5	8	9	10	17	17	20	23	28

Which one of the following statements is FALSE? (2分)

- ○ A. This problem can be solved by dynamic programming
- ○ B. The time complexity of this algorithm is $O(N^2)$
- ○ C. If $N \le M$, we have $R_N = \max\{P_N, \max_{1\le i<N}\{R_i + R_{N-i}\}\}$
- ◉ D. If $N > M$, we have $R_N = \max_{1\le i<N}\{R_i + R_{N-M}\}$

2-1 答案正确 (2 分) 创建提问

作者 陈越 单位 浙江大学

2-2 In dynamic programming, we derive a recurrence relation for the solution to one subproblem in terms of solutions to other subproblems. To turn this relation into a bottom up dynamic programming algorithm, we need an order to fill in the solution cells in a table, such that all needed subproblems are solved before solving a subproblem. Among the following relations, which one is impossible to be computed? (2分)

- ○ A. $A(i,j) = min(A(i-1,j), A(i,j-1), A(i-1,j-1))$
- ○ B. $A(i,j) = F(A(min\{i,j\}-1, min\{i,j\}-1), A(max\{i,j\}-1, max\{i,j\}-1))$
- ○ C. $A(i,j) = F(A(i,j-1), A(i-1,j-1), A(i-1,j+1))$
- ◉ D. $A(i,j) = F(A(i-2,j-2), A(i+2,j+2))$

2-2 答案正确 (2 分) 创建提问

作者 叶德仕 单位 浙江大学

2-3 Given a recurrence equation $f_{i,j,k} = f_{i,j+1,k} + \min_{0\le l\le k}\{f_{i-1,j,l} + w_{j,l}\}$. To solve this equation in an iterative way, we **cannot** fill up a table as follows: (2分)

- ○ A. `for k in 0 to n: for i in 0 to n: for j in n to 0`
- ◉ B. `for i in 0 to n: for j in 0 to n: for k in 0 to n`
- ○ C. `for i in 0 to n: for j in n to 0: for k in n to 0`
- ○ D. `for i in 0 to n: for j in n to 0: for k in 0 to n`

2-3 答案正确 (2 分) 创建提问

作者 杨欣豫 单位 浙江大学

Greedy Algorithm

1-1 Greedy algorithm works only if the local optimum is equal to the global optimum. (1分)

◉ T ○ F

1-1 答案正确 (1 分) 创建提问

作者 陈越 单位 浙江大学

1-2 Let S be the set of activities in Activity Selection Problem. Then the earliest finish activity a_m must be included in all the maximum-size subset of mutually compatible activities of S. (1分)

○ T ◉ F

1-2 答案正确 (1 分) 创建提问

作者 何钦铭 单位 浙江大学

NP

1-1 All decidable problems are NP problems. (1分)

○ T ◉ F

1-1 答案正确 (1 分) 创建提问

作者 陈越 单位 浙江大学

1-2 All NP problems are decidable. (1分)

◉ T ○ F

1-2 答案正确 (1 分) 创建提问

作者 陈越 单位 浙江大学

1-3 All NP-complete problems are NP problems. (1分)

◉ T ○ F

1-3 答案正确 (1 分) 创建提问

作者 陈越 单位 浙江大学

1-4 All NP problems can be solved in polynomial time in a non-deterministic machine. (1分)

◉ T ○ F

1-4 答案正确 (1 分) 创建提问

作者 陈越 单位 浙江大学

1-5 If a problem can be solved by dynamic programming, it must be solved in polynomial time. (2分)

○ T ◉ F

1-5 答案正确 (2 分) 创建提问

作者 陈越 单位 浙江大学

2-1 Among the following problems, __ is NOT an NP-complete problem. (2分)

- A. Vertex cover problem
- B. Hamiltonian cycle problem
- C. Halting problem
- D. Satisfiability problem

作者 陈越
单位 浙江大学

2-1 答案正确 (2 分) 创建提问

2-2 Suppose Q is a problem in NP, but not necessarily NP-complete. Which of the following is FALSE? (2分)

- A. A polynomial-time algorithm for SAT would necessarily imply a polynomial-time algorithm for Q.
- B. A polynomial-time algorithm for Q would necessarily imply a polynomial-time algorithm for SAT.
- C. If Q $\notin P$, then $P \neq NP$.
- D. If Q is NP-hard, then Q is NP-complete.

作者 叶德仕
单位 浙江大学

2-2 答案正确 (2 分) 创建提问