CS 260 Fundamentals of the Design and Analysis of Algorithms **Fall 2016**

HomeWork 6

Name: *Liangjian Chen*
ID: *#52006933* *November 15, 2016*

Problem 1 **Solution:**

(a) 99 - 100 - 99

(b) 100 - 1 - 1 - 100

(c) Denote $f[i][0]$ represents the maximum result of among first i nodes, node i is not chosen. $f[i][1]$ represents the maximum result of among first i nodes, node i is chosen. Then

$$f[i][j] = \begin{cases} 0 & i = 0 \\ max(f[i-1][1], f[i-1][0]) & i > 0 \text{ and } j = 0 \\ f[i-1][0] + w_i & i > 0 \text{ and } j = 1 \end{cases}$$

Final result is $\max\{f[n][0], f[n][1]\}$.

The state space is $O(n)$, transition function is $O(1)$. Total time is $O(n)$.

Problem 2 **Solution:**

(a) $\ell_1 = 0, \ell_2 = 100, \ell_3 = 1000$,$h_1 = 10, h_2 = 10, h_3 = 10$ Correct answer is choose h_1 and ℓ_3 whose sum is 1010. The algorithm will give ℓ_2 and h_3, whose sum is 110.

(b) Denote $f[i][0]$ represents the maximum result of among first i weeks, state week i is None. $f[i][1]$ represents the maximum result of among first i nodes, state of week i Occupied(either low or high stress). Then

$$f[i][j] = \begin{cases} 0 & i = 0 \\ max(f[i-1][1], f[i-1][0]) & i > 0 \text{ and } j = 0 \\ max(f[i-1][0] + h_i, f[i-1][1] + \ell_i) & i > 0 \text{ and } j = 1 \end{cases}$$

Final result is $\max\{f[n][0], f[n][1]\}$.

The state space is $O(n)$, transition function is $O(1)$. Total time is $O(n)$.

Problem 3 **Solution:**

(a) 6 nodes, 6 edges. (1,2),(2,5),(1,3),(3,4),(4,5),(5,6). Correct answer is $1 \to 3 \to 4 \to 5 \to 6$. Algorithm would find $1 \to 2 \to 5$.

(b) find the topological ordering of this graph stored in array A. Denote $f[i]$ as the longest path end at $A[i]$.

$$f[i][j] = \begin{cases} 0 & \text{no incoming edge} \\ max(f[j]) + 1 & \forall edge(j, A[i]) \end{cases}$$

Final result is $f[n]$.

The state space is $O(n)$, transition function is $O(1)$. Find the topological ordering is $O(n)$. Thus total time is $O(n)$.

Problem 4 **Solution:**

(a) $M = 100$,$N_1 = 1, N_2 = 10, N_3 = 10$,$h_1 = 10, h_2 = 1, h_3 = 1$. Correct answer is 12. The algorithm will give 103.

(b) $M = 0$, costs are given by the following table.

	Month 1	Month 2	Month 3	Month 4
NY	1000	0	1000	0
SF	0	1000	0	1000

Since the transport fee is 0, the we would choose the lower price every month which means locationmust be changed every month.

(c) Denote $f[i][0]$ represents the maximum result of among first i month, during month i you work in NY. $f[i][1]$ represents the maximum result of among first i month, during month i you work in SF.

$$f[i][j] = \begin{cases} 0 & i = 0 \\ max(f[i-1][1] + N_i + w, f[i-1][0] + N_i) & i > 0 \text{ and } j = 0 \\ max(f[i-1][1] + S_i, f[i-1][0] + S_i + w) & i > 0 \text{ and } j = 1 \end{cases}$$

Final result is $\max\{f[n][0], f[n][1]\}$.

The state space is $O(n)$, transition function is $O(1)$. Total time is $O(n)$.

Problem 5 **Solution:**

Denote $f[i]$ represents the maximum result of among first i characters.

$$f[i] = \begin{cases} 0 & i = 0 \\ \max\limits_{0 \le j \le i-1} \{f[j] + quality(y_{j+1}y_{j+2}...y_i)\} & i \ge 1 \end{cases}$$

Final result is $f[n]$.

The state space is $O(n)$, transition function is $O(1)$. Total time is $O(n)$.

Problem 6 **Solution:**

Define, $S[i]$ as the sum of first i items' length.

$$S[i] = \begin{cases} 0 & i = 0 \\ S[i-1] + C[i] & i \ge 1 \end{cases}$$

Then, denote $f[i]$, is the minimum summation of square error.

$$f[i] = \begin{cases} 0 & i = 0 \\ \min\limits_{0 \le j \le i-1 \text{ and } L \ge S[i] - S[j] + i - j - 1} \{f[j] + (L - (S[i] - S[j] + i - j - 1))^2\} & i \ge 1 \end{cases}$$

Final result is $f[n]$.

The state space is $O(n)$, transition function is $O(n)$. Total time is $O(n^2)$.

Problem 7 **Solution:**

Define, $M[i]$ as the maximum price from i^{th} day to the last day.

$$M[i] = \begin{cases} p(n) & i = n \\ \max\{M[i+1], p(i)\} & i \leq n-1 \end{cases}$$

Define, $Day[i]$ as the maximum price date from i^{th} day to the last day.

$$Day[i] = \begin{cases} n & i = n \\ Day[i+1] & M[i] = M[i+1], i \leq n-1 \\ i & M[i] > M[i+1], i \leq n-1 \end{cases}$$

Set variable Ans as the maximum profit we can make.
iterator all price. If $M[i]-p(i)$ larger than current answer, the update the answer, and recode i and $Day[i]$ as the optimal pair (i, j). The state space is $O(n)$, transition function is $O(1)$. Total time is $O(n)$.

Problem 8 **Solution:**

(a) the instance is shown below.

	1	2	3	4
x_i	2	2	2	2
$f(i)$	1	1	1	2

Correct answer is 4, which shoot laser every day. However the algorithm will give 2 which is just shooting laser in 4^{th} day.

(b) Denote $dp[i]$ as the maximum number of robot killed in first i days, and we use the laser in i^th day.

$$dp[i] = \begin{cases} 0 & i = 0 \\ \max\limits_{1 \leq j \leq i-1} \{f[j] + \min\{x_i, f(i-j)\}\} & i \leq n-1 \end{cases}$$

Final result is $\max\limits_{1 \leq i \leq n} f[i]$.

The state space is $O(n)$, transition function is $O(n)$. Total time is $O(n^2)$.

Problem 9 **Solution:**

(a)

	Day 1	Day 2	Day 3	Day 4	Day 5
x	1001	1001	1001	1001	1001
s	1000	4	3	2	1

Optimal solution is reboot in Day 2 and Day 4, work on Day 1 ,3 ,5.

(b) Denote $dp[i][j]$ as the maximum number of terabytes processed i days and computer has been continually working for j days.

$$dp[i][j] = \begin{cases} 0 & i = 0 \\ \max\limits_{1 \leq j \leq i} \{f[i-1][j]\} & i \geq 1, j = 0 \\ f[i-1][j-1] + \min\{x_i, s[j]\} & i \geq 1, j \geq 1 \end{cases}$$

. Also, when we process the dp, we need use $pre[i]$ to records which j update the $dp[i][0], 1 \le i \le n$ i.e. $pre[i] = argmax_{1\le j\le i-1}\{dp[i-1][j]\}$.

Then, first we find the maximum data we processed in n^{th} day and record its corresponding j. So from $n-j+1$ day to n^{th} day, it is works and $n-j$ day is reboot. Then we use $pre[n-j]$ to find how many days computer continually works in this period of time. By using the method shows above again and again until we meet the first day. Now we have construct the optimal solution.

Problem 10 **Solution:**

(a) The instance is shown below:

	Day 1	Day 2
x	1	1000
s	2	1

Correct answer is 1001, but the algorithm will find 3.

(b) Denote $f[i][0]$ represents the maximum result of among first i days, and using computer A in i^{th} day. $f[i][1]$ represents the maximum result of among first i days, and using computer B in i^{th} day. Then

$$dp[i][j] = \begin{cases} a_1 & i=1, j=0 \\ b_1, & i=1, j=1 \\ \max\{f[i-1][0]+a_i, f[i-2][1]+b_i\} & i\ge 2, j=0 \\ \max\{f[i-1][1]+b_i, f[i-2][0]+a_i\} & i \ge 2, j=1 \end{cases}$$

Final result is $\max\{f[n][0], f[n][1]\}$.

The state space is $O(n)$, transition function is $O(1)$. Total time is $O(n)$.