50.004 – Homework 12
Ashlyn Goh 1002840 Cl02 Nov 2018

Q1.

(a) To compute the longest common subsequence for the strings TGTCGA and GCAGCTTGCC:

	j	0	1	2	3	4	5	6	7	8	9	10
i		Y_j	G	C	A	G	C	T	T	G	C	C
0	X_i	0	0	0	0	0	0	0	0	0	0	0
1	T	0	↑ 0	↑ 0	↑ 0	↑ 0	↑ 0	↖1	↖1	← 1	← 1	← 1
2	G	0	↖1	← 1	← 1	↖1	← 1	↑ 1	↑ 1	↖2	← 2	← 2
3	T	0	↑ 1	↑ 1	↑ 1	↑ 1	↑ 1	↖2	↖2	↑ 2	↑ 2	↑ 2
4	C	0	↑ 1	↖2	← 2	← 2	↖2	↑ 2	↑ 2	↑ 2	↖3	↖3
5	G	0	↖1	↑ 2	↑ 2	↖3	← 3	← 3	← 3	↖3	↑ 3	↑ 3
6	A	0	↑ 1	↑ 2	↖3	↑ 3	↑ 3	↑ 3	↑ 3	↑ 3	↑ 3	↑ 3

From the table above, the longest common subsequence computed is TGC.

(b) No, the answer is not unique. Two other LCS for the two given strings are TTC and TTG.

Q2.

Items we have:

Item No.	**Size**	**Value**	**Value/Size**
1	2	1	0.50
2	2	2	1.00
3	4	5	1.25
4	5	6	1.20

Assuming this is a **0/1 Knapsack** and multiple copies of an item is **NOT ALLOWED**.

	S=0	**1**	**2**	**3**	**4**	**5**	**6**
Item=0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1
2	0	0	2	2	3	3	3
3	0	0	2	2	5	5	7
4	0	0	2	2	5	6	**7**

Hence, the solution to this problem is value 7, where items 2 and 3 will be taken.

(a) The elements in the second last column are 0, 1, 3, 5, 6.

(b) True. In the Greedy approach, we would choose to take the item with the most value-to-size ratio at each stage. Thus, we would end up with item 3 and then item 2; and thus, a value of 7 (similar to the answer given by DP). In this question, the greedy approach does give us an optimal solution and since a greedy approach is usually much faster than that of Dynamic Programming, the greedy approach would yield a better performance. (However, the greedy approach would not always work for 0/1 knapsack questions if we are unable to fill the knapsack to capacity. The empty space would lower the effective value per size of our load, leading to a less than optimal solution than that provided by DP.)

50.004 – Homework 12
Ashlyn Goh 1002840 Cl02 Nov 2018

Q3 DP recurrence relation:

$m[i,j] \rightarrow$ costs of optimal solutions to subproblems.

$$m[i,j] = \begin{cases} 0 & \text{if } i=j \\ \min\limits_{i \le k < j} \{m[i,k] + m[k+1,j] + p_{i-1}p_k p_j\} & \text{if } i<j \end{cases}$$

$P_0 = 4$ $P_1=1$ $P_2=3$ $P_3=2$ $P_4=3$

DP Table:

i \ j	**1**	**2**	**3**	**4**
1	**0**	**12**	**14**	**24**
2	X	**0**	**6**	**12**
3	X	X	**0**	**18**
4	X	X	X	**0**

(k=1) $M[1,2] = M[1,1] + M[2,2] + P_0P_1P_2 = 0 + 0 + 4(1)(3) = 12$

(k=2) $M[2,3] = M[2,2] + M[3,3] + P_1P_2P_3 = 0 + 0 + 1(3)(2) = 6$

(k=3) $M[3,4] = M[3,3] + M[4,4] + P_2P_3P_4 = 0 + 0 + 3(2)(3) = 18$

$M[1,3]$: (k=1) -> $M[1,1] + M[2,3] + P_0P_1P_3 = 0 + 6 + 4(1)(2) = 14$ (the min value)

(k=2) -> $M[1,2] + M[3,3] + P_0P_2P_3 = 12 + 0 + 4(3)(2) = 36$

$M[2,4]$: (k=2) -> $M[2,2] + M[3,4] + P_1P_2P_4 = 0 + 18 + 1(3)(3) = 27$

(k=3) -> $M[2,3] + M[4,4] + P_1P_3P_4 = 6 + 0 + 1(2)(3) = 12$ (the min value)

$M[1,4]$: (k=1) -> $M[1,1] + M[2,4] + P_0P_1P_4 = 0 + 12 + 4(1)(3) = 24$ (the min value)

(k=2) -> $M[1,2] + M[3,4] + P_0P_2P_4 = 12 + 18 + 4(3)(3) = 66$

(k=3) -> $M[1,3] + M[4,4] + P_0P_3P_4 = 14 + 0 + 4(2)(3) = 38$

Minimum cost to perform the matrix multiplication is 24.
Order of multiplication: $(A_1)((A_2\,A_3)\,A_4)$

To check:

$(4\times1)\left[\left((1\times3)(3\times2)\right)(2\times3)\right]$

$(4\times1)\left[\left(\overset{6}{1\times2}\right)(2\times3)\right]$

$(4\times1)\left[\overset{6}{1\times3}\right]$

$\overset{12}{4\times3}$

Total cost: $6+6+12 = 24$