CMPE/CISC-365*
Quiz #3
November 8, 2019

Student Number (Required) ______________________

Name (Optional)________________________________

This is a closed book test. You may refer to one 8.5 x 11 data sheet.

This is a 50 minute test.

Please write your answers in ink. Pencil answers will be marked, **but will not be re-marked under any circumstances.**

The test will be marked out of 50.

Question 1	/24
Question 2	/24
Question 3	/2
TOTAL	/50

"I guess the issue for me is to keep things dynamic."

— Robert Downey Jr.

QUESTION 1 (24 Marks)

You have been chosen to plan a canoe trip down the NottaLottaWatta River for the Queen's University Environmental Exploration Nature Society (acronym: QUEENS) . Canoes are available for rent at trading posts along the river. You will start the trip by renting a canoe at Post 1 (where the river begins) and end the trip in Post n (the end of the river). BUT ... you don't have to keep the same canoe the whole way. You can stop at any post, drop off the canoe you have and rent another one. You can only travel downstream. For all pairs (a, b) with $a < b$, the cost of renting a canoe at Post a and dropping it off at Post b is given by a predetermined matrix Cost(a, b).

For example if there are five posts in total, the costs might be

Cost(a,b) matrix		Post b				
		P_1	P_2	P_3	P_4	P_5
Post a	P_1	x	10	35	50	65
	P_2	x	x	30	35	45
	P_3	x	x	x	15	25
	P_4	x	x	x	x	20
	P_5	x	x	x	x	x

In the example shown, you could rent a canoe from P_1 to P_2, then rent another from P_2 to P_3, then another from P_3 to P_4, then another from P_4 to P_5. This would cost $10 + 30 + 15 + 20 = 75$. Another solution would be to rent a canoe from P_1 to P_3 (cost 35) and another canoe from P_3 to P_5 (cost 25) with a total cost of 60.

Your job is to plan the sequence of canoe rentals to **minimize the total cost.**

We can think of the problem like this: We have to return our last canoe at P_n. We could have rented that canoe at any of $P_1, P_2, \ldots P_{n-1}$. Where-ever we rented our last canoe, we have to solve the rest of the trip optimally from P_1 to that point.

(a) **[6 marks]** How many different possible solutions are there? Remember there are n Posts, where n can be any integer ≥ 2. Explain your answer.

Solution: We have to rent a canoe at P_1, and we can also rent canoes at any subset of $\{P_2, \ldots, P_{n-1}\}$. Thus the number of possible solutions is the number of subsets of $\{P_2, \ldots, P_{n-1}\}$ ... which is 2^{n-2}

Marking:

Correct answer with explanation	**6 marks**
Correct answer without explanation	**4 marks**
"Close" incorrect answer (such as 2^{n-1}) with explanation (such as "any subset")	**3 marks**
"Close" incorrect answer without explanation	**2 marks**
"Wayout" answer (such as n) with or without explanation	**1 mark**

(b) **[12 marks]** Let $MC(i)$ = the minimum cost of getting from P_1 to P_i

(so $MC(n)$ is our over-all solution)

Give a complete statement of a recurrence relation for $MC(i)$.

As a starting point, here is a base case: $MC(2) = Cost(1,2)$

Solution:

for all i > 2:

MC(i) = min (Cost(1,i),
MC(i-1) + Cost(i-1,i),
MC(i-2) + Cost(i-2,i),
MC(i-3) + Cost(i-3,i),
…
MC(2) + Cost(2,i)
)

Marking:

The hint should suggest that the cost of getting to P_i = the cost of the final canoe that gets us there, plus the minimum cost of getting to the post where we rent that canoe.

The key concept is that the value of MC(i) depends on *all* the previous values.

A student whose answer captures these ideas should get at least 8/12 even if they are unable to correctly express the recurrence relation. Giving 10/12 or 11/12 is appropriate if the answer is close to being correct.

A student whose answer shows that they understand the concept and purpose of a recurrence relation, but not how to create one for this problem, should get at least 6/12

A student whose answer shows only a weak understanding of recurrence relations should get about 3/12

(c) **[6 marks]** Determine the computational complexity of using a Dynamic Programming approach to solve this problem. Explain your answer.

Solution: Using the recurrence relation given, the value of MC(i) is computed by taking the min of i-1 values, each of which is computed in constant time. The sum of all computations for MC(n) is thus proportional to the sum 1 + 2 + … + n-1, which is in $O(n^2)$

Marking:

Same rubric as part (a)

QUESTION 2 (24 Marks)

You have landed a job in a steel mill. The mill produces steel bars of random lengths (all lengths are integers). Strangely, customers seem to prefer steel bars of regular lengths. Your job is cut the raw steel bars into shorter lengths in the most profitable way.

More precisely, you need to cut a bar that is n metres long into shorter pieces, each piece being ≤ 5 metres long. Each short piece has a profit value to the company as shown in this table:

Length	1	2	3	4	5
Profit	2	3	6	9	11

So if n = 6, you could cut the bar into a piece of length 5 and a piece of length 1, with a total profit of 13 … or you could cut the bar into a piece of length 4 and a piece of length 2, with a total profit of 12. There are many other possibilities, including cutting the bar into six pieces of length 1, or two pieces of length 2 and two pieces of length 1, etc.

But if n = 7, cutting a piece of length 5 and a piece of length 2 gives a total profit of 14, while a piece of length 4 and a piece of length 3 gives a total profit of 15. You could also cut the bar into two pieces of size 2 and one piece of size 3, etc. etc.

Design a Dynamic Programming algorithm to find the **maximum profit** obtainable from a bar of length n, where n can be any positive integer.

Hint: remember the dynamic programming algorithm for change-making.

(a) Design a recurrence relation for MaxProfit(n), including base case(s) and a recursive part **[8 marks]**

Solution: for each of the lengths between 1 and 5 except 2, the profit cannot be improved by cutting. For a length of 2, we get a better profit (4) by cutting it into two pieces of size 1. So the base cases are:
MaxProfit(n) = Profit(n)   for n = 1, 3, 4, 5
MaxProfit(2) = 4

For n $\geq$ 6, the recurrence relation is:

$$\mathit{MaxProfit}(n) = \max\big(\ 2 + \mathit{MaxProfit}(n-1),\ 3 + \mathit{MaxProfit}(n-2),\ 6 + \mathit{MaxProfit}(n-3),\ 9 + \mathit{MaxProfit}(n-4),\ 11 + \mathit{MaxProfit}(n-5)\ \big)$$

(*Note that we can actually leave out the 3+MaxProfit(n-2) option since it will never be optimal … but it's ok to leave it in.*)

Marking:

Base Cases: **3 marks**

Recursive Part: **5 marks**

As with the recurrence relation part of the previous question, please give part marks if the student understands what is to be done but has some errors in their solution.

(b) Specify how you will store information **[5 marks]**

Solution: ***Since the recurrence relation has only one parameter, we can store information in a 1-dimensional array.***

Marking:

Students might suggest storing the results in a hash-table – it really offers no advantage since we need to solve all the subproblems up to n anyway. I would give 4 marks for this – it's overkill.

Students might also suggest using a 2-dimensional array (I'm not sure how!) - I would give 3 marks for this.

If a student's answer shows that they really didn't understand the concept of storing the results of subproblems in an easily-accessible way, they should get 1 mark for trying.

(c) Specify how you will order your computations **[5 marks]**

***Solution:** MaxProfit(n) depends only on values of MaxProfit(x) where x < n. We can perform the computations from MaxProfit(1) up to MaxProfit(n) – this ensures that all information needed for each MaxProfit value is available when it is needed.*

Marking:

Students may also suggest working from the top down (recursively) and storing each value the first time the subproblem is encountered, then looking the values up on subsequent requests. This is ok – it has the same complexity (just a bit more overhead).

If a student's answer shows that they understand the question but they cannot relate it to this problem, they should get about 2 or 3 out of 5.

(d) Explain how you will reconstruct the set of cuts from the computed MaxProfit(n) information **[6 marks]**

Solution: ***Once we know the value of MaxProfit(n), we can look at its five possible predecessors (the values for n-1, n-2, n-3, n-4 and n-5) and determine which cut length resulted in the maximum value. This tells us what the final cut was. We work back in this manner to find all the cuts.***

Marking:

Students might also suggest "carrying" the optimal set of cuts along in the table, so the solution would be immediately available, or carrying some "most recent cut" information along in which case the solution details can be reconstructed without doing any comparisons. These are both completely acceptable.

An answer which is fundamentally correct but contains some errors should get at least 4/6

If the student's answer shows they understand what is being asked but they can't express a solution for this problem, they should get 2 or 3 out of 6.

QUESTION 3 (2 Marks)

True or **False**:

The 2018 Award for Excellence in Dynamic Programming was won by Netflix.

TRUE FALSE

Solution: False

Marking: 2 marks for everyone