

CSCI 3104 PS9b

Jonathan Phouminh

TOTAL POINTS

45 / 51

QUESTION 1

21 pts

1.1 5 / 5

- ✓ + **1.5 pts** Part (i) correct
- ✓ + **1.5 pts** Part (ii) correct.
- ✓ + **2 pts** Correct recurrence
 - + **1 pts** Base case missing in part (i)
 - **0.5 pts** Missing return statement in Part (i)
 - + **1 pts** Partially correct recurrence
 - + **1 pts** Syntax errors in (i)
 - **5 pts** Plagiarism
 - + **0 pts** Incorrect tree of function calls for P5 in (ii).
 - + **0 pts** Incorrect recurrence relation for (iii)
 - + **1 pts** Partially correct tree for (ii)
 - + **0 pts** Incorrect part (i) answer
- ☹ You should label the nodes of the tree diagram with the calls, rather than the return values. This makes it easier for us to follow your work and grade it.
We don't make two or more calls to $P(n-1)$, so the recurrence should be $T(n) = T(n-1) + T(n-2)$.

1.2 6 / 6

- ✓ + **1.5 pts** Part (i) correct
- ✓ + **1.5 pts** Part (ii) correct
- ✓ + **1.5 pts** Part (iii) correct
- ✓ + **1.5 pts** Part (iv) correct
 - + **1 pts** Switched order of P_0 and P_1 for (iii) (Or order unspecified for $n=0,1$)
 - + **1 pts** Minor errors in (i)
 - **6 pts** Plagiarism
 - + **1 pts** Partially correct tree of function calls (ii)
 - + **0 pts** Incorrect time complexity and recurrence (iv)

- + **0 pts** Incorrect pseudocode (i)
- + **0 pts** Incorrect or incomplete tree of function calls (ii)
 - **0.5 pts** Missing or incorrect justification for (iv)
 - + **0 pts** Incorrect order (iii)
 - + **0.5 pts** Several or significant errors in the pseudocode (i)
 - + **1 pts** Partially correct order (iii)
- ① Here, $P(3)$ calls $P(1)$ also.

1.3 5 / 5

- ✓ + **2 pts** (i) Correct
- ✓ + **2 pts** (ii) Correct
- ✓ + **1 pts** (iii) Correct
 - + **0 pts** Plagiarism
 - + **0 pts** (i) Your algorithm is neither iterative nor bottom up.
 - + **0.5 pts** (iii) Included either correct time or space complexity, but not both
 - + **1 pts** You switched the order in which $P[0]$ and $P[1]$ are being filled.
 - + **0.5 pts** (i) you are using recursive implementation, instead of dp based iterative bottom up implementation.
 - + **0 pts** not attempted
 - + **1 pts** (ii) The ordering is correct but the values are incorrect.

1.4 3 / 3

- ✓ + **3 pts** Correct
 - **3 pts** Plagiarism
 - + **1 pts** Algorithm does not have $\Theta(1)$ space usage.
 - + **0 pts** No justification that the algorithm has the correct space complexity.

- + **1.5 pts** Correct algorithm
- + **0 pts** Incorrect algorithm
- **0.2 pts** Minor mistake
- + **0 pts** not attempted

1.5 2 / 2

- ✓ + **2 pts** Correct
- **2 pts** Plagiarism
- + **1 pts** Correct asymptotics
- + **0.5 pts** Partially correct explanation
- + **0 pts** No discussion
- + **1 pts** Sufficient explanation
- + **0.5 pts** Partially correct asymptotics
- **0.2 pts** Minor errors
- + **0 pts** not attempted

QUESTION 2

10 pts

2.1 2 / 2

- + **0 pts** Incorrect or no solution provided. Please refer to the solution file.
- **2 pts** Plagiarism
- ✓ + **1 pts** Correct example considered
- ✓ + **1 pts** Work shown to prove the greedy strategy fails on the example considered
- + **0.5 pts** Did not provide explicit constructions

2.2 4 / 4

- ✓ + **4 pts** Correct and all conditions satisfied
- + **0 pts** Plagiarism
- + **1 pts** Correct base cases considered till when $n = 2$
- + **1 pts** Correct recursion formula used
- + **1 pts** $O(n)$ space complexity used
- + **1 pts** Passed all the random test cases
- + **0 pts** Empty or Incorrect solution or code file not found

2.3 4 / 4

- ✓ + **4 pts** Correct and all conditions satisfied.
- **4 pts** Plagiarism

- + **1 pts** Correct bases considered till when $n = 2$
- + **1 pts** Correct recursion formula used
- + **1 pts** $O(1)$ space complexity used
- + **1 pts** Passed all the random test cases
- + **0 pts** Incorrect or Empty solution or code file not found

QUESTION 3

3 6 / 10

- + **10 pts** Correct. PS: Should write down your transition very clear. As this is the most important part for DP problem.
- + **3 pts** The transition function is correct.
- + **2 pts** The algorithm works correctly.
- ✓ + **3 pts** Transition function and algorithm most correct, but still some problems, please check the solution carefully.
- + **3 pts** Analyze decision of those sub-problems correctly.
- + **1 pts** Analyze the time complexity correctly.
- ✓ + **1 pts** Analyze the space complexity correctly.
- + **0 pts** Incorrect/Not attempted.
- + **2 pts** Doesn't explain reasons well, but give a reasonable definition for dp array/matrix.
- ✓ + **2 pts** Some correct analysis for analyzing sub-problems
- **10 pts** Plagiarism

QUESTION 4

10 pts

4.1 6 / 6

- ✓ + **2.5 pts** Correct values across table
- ✓ + **2 pts** Correct paths between all table values
- ✓ + **1.5 pts** Correct backtrack path and LCS values
- **1 pts** Errors in table values
- **0.5 pts** Minor errors in paths
- **1 pts** Errors in backtrack path.
- **1 pts** Wrong LCS value and/or length
- + **0 pts** Plagiarism
- + **0 pts** No answer

4.2 2 / 4

+ 4 pts Correct tree

+ 3 pts Right sub tree's children missing (children of
LCS(FUB, B))

✓ + 2 pts Partially correct tree

+ 0 pts Plagiarism

+ 0 pts No answer

+ 0 pts Incorrect

1.1 5 / 5

✓ + 1.5 pts Part (i) correct

✓ + 1.5 pts Part (ii) correct.

✓ + 2 pts Correct recurrence

+ 1 pts Base case missing in part (i)

- 0.5 pts Missing return statement in Part (i)

+ 1 pts Partially correct recurrence

+ 1 pts Syntax errors in (i)

- 5 pts Plagiarism

+ 0 pts Incorrect tree of function calls for P5 in (ii).

+ 0 pts Incorrect recurrence relation for (iii)

+ 1 pts Partially correct tree for (ii)

+ 0 pts Incorrect part (i) answer

💬 You should label the nodes of the tree diagram with the calls, rather than the return values. This makes it easier for us to follow your work and grade it.

We don't make two or more calls to $P(n-1)$, so the recurrence should be $T(n) = T(n-1) + T(n-2)$.

1.2 6 / 6

- ✓ + **1.5 pts** Part (i) correct
- ✓ + **1.5 pts** Part (ii) correct
- ✓ + **1.5 pts** Part (iii) correct
- ✓ + **1.5 pts** Part (iv) correct
 - + **1 pts** Switched order of P_0 and P_1 for (iii) (Or order unspecified for n=0,1)
 - + **1 pts** Minor errors in (i)
 - **6 pts** Plagiarism
 - + **1 pts** Partially correct tree of function calls (ii)
 - + **0 pts** Incorrect time complexity and recurrence (iv)
 - + **0 pts** Incorrect psuedocode (i)
 - + **0 pts** Incorrect or incomplete tree of function calls (ii)
 - **0.5 pts** Missing or incorrect justification for (iv)
 - + **0 pts** Incorrect order (iii)
 - + **0.5 pts** Several or significant errors in the pseudocode (i)
 - + **1 pts** Partially correct order (iii)

1 Here, P(3) calls P(1) also.

1.3 5 / 5

✓ + 2 pts (i) Correct

✓ + 2 pts (ii) Correct

✓ + 1 pts (iii) Correct

+ 0 pts Plagiarism

+ 0 pts (i) Your algorithm is neither iterative nor bottom up.

+ 0.5 pts (iii) Included either correct time or space complexity, but not both

+ 1 pts You switched the order in which $P[0]$ and $P[1]$ are being filled.

+ 0.5 pts (i) you are using recursive implementation, instead of dp based iterative bottom up implementation.

+ 0 pts not attempted

+ 1 pts (ii) The ordering is correct but the values are incorrect.

1.4 3 / 3

✓ + 3 pts Correct

- 3 pts Plagiarism

+ 1 pts Algorithm does not have $\Theta(1)$ space usage.

+ 0 pts No justification that the algorithm has the correct space complexity.

+ 1.5 pts Correct algorithm

+ 0 pts Incorrect algorithm

- 0.2 pts Minor mistake

+ 0 pts not attempted

1.5 2 / 2

✓ + 2 pts Correct

- 2 pts Plagiarism

+ 1 pts Correct asymptotics

+ 0.5 pts Partially correct explanation

+ 0 pts No discussion

+ 1 pts Sufficient explanation

+ 0.5 pts Partially correct asymptotics

- 0.2 pts Minor errors

+ 0 pts not attempted

2.1 2 / 2

- + **0 pts** Incorrect or no solution provided. Please refer to the solution file.
- **2 pts** Plagiarism
- ✓ + **1 pts** Correct example considered
- ✓ + **1 pts** Work shown to prove the greedy strategy fails on the example considered
- + **0.5 pts** Did not provide explicit constructions

2.2 4 / 4

✓ + **4 pts** Correct and all conditions satisfied

+ **0 pts** Plagiarism

+ **1 pts** Correct base cases considered till when $n = 2$

+ **1 pts** Correct recursion formula used

+ **1 pts** $O(n)$ space complexity used

+ **1 pts** Passed all the random test cases

+ **0 pts** Empty or Incorrect solution or code file not found

2.3 4 / 4

✓ + 4 pts Correct and all conditions satisfied.

- 4 pts Plagiarism

+ 1 pts Correct bases considered till when $n = 2$

+ 1 pts Correct recursion formula used

+ 1 pts $O(1)$ space complexity used

+ 1 pts Passed all the random test cases

+ 0 pts Incorrect or Empty solution or code file not found

3 6 / 10

+ **10 pts** Correct. PS: Should write down your transition very clear. As this is the most important part for DP problem.

+ **3 pts** The transition function is correct.

+ **2 pts** The algorithm works correctly.

✓ + **3 pts** Transition function and algorithm most correct, but still some problems, please check the solution carefully.

+ **3 pts** Analyze decision of those sub-problems correctly.

+ **1 pts** Analyze the time complexity correctly.

✓ + **1 pts** Analyze the space complexity correctly.

+ **0 pts** Incorrect/Not attempted.

+ **2 pts** Doesn't explain reasons well, but give a reasonable definition for dp array/matrix.

✓ + **2 pts** Some correct analysis for analyzing sub-problems

- **10 pts** Plagiarism

4.1 6 / 6

- ✓ + **2.5 pts** Correct values across table
- ✓ + **2 pts** Correct paths between all table values
- ✓ + **1.5 pts** Correct backtrack path and LCS values
 - **1 pts** Errors in table values
 - **0.5 pts** Minor errors in paths
 - **1 pts** Errors in backtrack path.
 - **1 pts** Wrong LCS value and/or length
- + **0 pts** Plagiarism
- + **0 pts** No answer

4.2 2 / 4

+ 4 pts Correct tree

+ 3 pts Right sub tree's children missing (children of $\text{LCS}(\text{FUB}, \text{B})$)

✓ + 2 pts Partially correct tree

+ 0 pts Plagiarism

+ 0 pts No answer

+ 0 pts Incorrect