Dynamic Programming II

Reset Progress | Reveal Solutions

1 Longest Common Subsequence Forensics

We are computing the longest common subsequence between two strings of length four $S = X_1X_2X_3X_4$ and $T = Y_1Y_2Y_3Y_4$. We fill the array C where $C_{i,j}$ is the length of the longest common subsequence between the prefix of length i from S and the prefix of length j from T. The array C can be found below with some entries masked:

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	♡
4	0	1	♣	♠	♢

What can be said about X_1 and Y_4?
- (●) They are equal.
- ( ) They are different.
- ( ) They could be equal or different.

Correct

What can be said about X_2 and Y_3?
- (●) They are equal.
- ( ) They are different.
- ( ) They could be equal or different.

Correct

What can be said about X_2 and Y_4?
- ( ) They are equal.
- (●) They are different.
- ( ) They could be equal or different.

Correct

What is the value of ♡?
- ( ) 0
- (●) 1
- ( ) 2
- ( ) 3
- ( ) 4
- ( ) Multiple answers could be correct.

Correct

What is the value of ♠?
- ( ) 0
- (●) 1
- ( ) 2
- ( ) 3
- ( ) 4
- ( ) Multiple answers could be correct.

Correct

What is the value of ♣?
- ( ) 0
- (●) 1
- ( ) 2
- ( ) 3
- ( ) 4
- ( ) Multiple answers could be correct.

Correct

What is the value of ♢?
- ( ) 0
- (●) 1
- ( ) 2
- ( ) 3
- ( ) 4
- ( ) Multiple answers could be correct.

Correct

Suppose that in some (possibly different) instance of the longest common subsequence problem, we have $C_{i,j} = C_{i-1,j-1} + 1$. Does that necessarily mean the i-th character of the first string and the j-th character of the second string are equal?
- ( ) Yes
- (●) No

Correct

2 LCS Space Complexity

Consider the LCS problem from lecture 13 and our dynamic programming algorithm for it. Given input stings of lengths m and n, what is the memory complexity of this algorithm?
- ( ) $O(n + m)$
- ( ) $O((n + m)^2)$
- (●) $O(mn)$

Correct

When we are filling up the i-th row of our dynamic programming table C, what rows do we need to have access to?
- ( ) We need to access all the n rows.
- ( ) We need to access the first i rows.
- (●) We need to access the values in the i-th row and $(i - 1)$-th row.

Correct

Given the observation above can we optimize our space Complexity further?
- ( ) No, the best memory Complexity is $O(nm)$.
- ( ) Yes we can reduce the memory complexity to $O(n \log(m))$.
- ( ) Yes we can reduce the memory complexity to $O(\frac{m}{n})$.
- (●) Yes we can reduce the memory complexity to $O(\min(m, n))$.

Correct

3 Knapsack Forensics

Suppose we are trying to solve an instance of the unbounded knapsack problem. We fill the array K whose entry K_i gives us the maximum value we can obtain from a knapsack of capacity i.

0	1	2	3	4	5	6
0	0	1	3	♠	♡	♢

What is the minimum possible value for ♠?

3

Correct

What is the minimum possible value for ♡?

4

Correct

What is the minimum possible value for ♢?

6

Correct

If we fill a knapsack of capacity 3 optimally, how many items do we put in the knapsack?
- ( ) 0
- (●) 1
- ( ) 2
- ( ) 3
- ( ) Multiple answers could be correct.

Correct

4 Maximum Independent Set on a Tree

Consider the maximum independent on trees problem from the lecture 13 slides. We saw a top-down dynamic programming approach to solve this problem. Now we'd like to see how a bottom up approach to solve MIS on a tree would look like. Which one of the following statements is correct?
- ( ) In order to solve this problem bottom-up we need to order the vertices by increasing DFS finish time.
- ( ) In order to solve this problem bottom-up we need to order the vertices by decreasing DFS start time.
- (●) Both of the above.
- ( ) Any ordering would work.

Correct

What is the best run-time for the bottom-up approach to solve the MIS on a tree problem?
- (●) $O(n + m)$
- ( ) $O((n + m) \log(n))$
- ( ) $O((n + m)^2)$

Correct