<u>Dashboard</u> / My courses / <u>Information Technology</u> / <u>IT300 - 26722</u> / Week 14: 26 October - 30 October / <u>Theory</u>	es / <u>Information Technology</u> / <u>IT300 - 26722</u> / Week 14: 26 October - 30 October / <u>Theory Quiz 2</u>
---	---

	Finished
-	Saturday, 31 October 2020, 11:30 AM
	29 mins 35 secs 6.00 out of 20.00 (30 %)
uestion 1	
orrect	
ark 1.00 out of 1.00	
	he six letters p,q,r,s,t,u in a file is 1/32, 1/32, 1/16, 1/8, 1/4 and 1/2 respectively. Out of the following codes, which an coding for the letters?
Select one or more	
0000, 0001,001	,01,10,11
00000, 00001, 0	0001, 001, 01, 1 🗸
✓ 11111, 11110,	1110, 110, 10, 0 🗸
000, 001, 010, 0	11, 10, 11
The correct answers	ect. s are: 11111, 11110, 1110, 110, 10, 0, 00000, 00001, 0001, 001, 01,
ruestion 2 ncorrect	
The correct answers Question 2 Accorrect Mark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight	(the maximum value that can be picked) of the Knapsack problem on the following input? 10 16 16 6 16 ghts of the four items are given in the V and W arrays above (item <i>i</i> has value V[i] and weight W[i]). Total capacity
The correct answers Question 2 Incorrect Mark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1	(the maximum value that can be picked) of the Knapsack problem on the following input? 20 16 8 6 ghts of the four items are given in the V and W arrays above (item <i>i</i> has value V[i] and weight W[i]). Total capacity 2 units. You can only pick the entire item, not fractions of it.
The correct answers Tuestion 2 The correct Thank 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more	(the maximum value that can be picked) of the Knapsack problem on the following input? 20 16 8 6 ghts of the four items are given in the V and W arrays above (item <i>i</i> has value V[i] and weight W[i]). Total capacity 2 units. You can only pick the entire item, not fractions of it.
The correct answers Tuestion 2 Tuestion 2 The correct Tark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more 26 ×	(the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value tha
The correct answers uestion 2 correct lark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more 26 × None of the ot	(the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value tha
The correct answers Question 2 Accorrect Mark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more 26 × None of the ot	(the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value tha
The correct answers uestion 2 correct lark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more 26 × None of the ot 28 32	(the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value tha
The correct answers uestion 2 correct lark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more 26 × None of the ot	(the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value that can be picked) of the Knapsack problem on the following input? (the maximum value tha
The correct answers uestion 2 correct lark 0.00 out of 1.00 What is the output V 12 10 2 W 5 3 The values and weight of the knapsack is 1 Select one or more 26 × None of the ot 28 32	(the maximum value that can be picked) of the Knapsack problem on the following input? (20 16 8 6 ghts of the four items are given in the V and W arrays above (item <i>i</i> has value V[i] and weight W[i]). Total capacity 2 units. You can only pick the entire item, not fractions of it.

1 of 7

Theory Quiz 2: Attempt review

Question $\bf 3$

Correct

Mark 2.00 out of 2.00

Suppose S(n) represents the total number of possible binary strings of length n which have two successive 1s. Then which of the following is true?

Select one or more:

$$\ \ \square \ \ S(n) = 2^{n-2} + 2S(n-1) + S(n-2)$$

$$S(n) = 2^{n-2} + S(n-1) + S(n-2)$$

~

$$\Box S(n) = 2^{n-2} + S(n-1) + 2S(n-2)$$

$$\Box S(n) = 2^{n-2} + 2S(n-1) + 2S(n-2)$$

□ None of the others

Your answer is correct.

The correct answer is: $S(n) = 2^{n-2} + S(n-1) + S(n-2)$

Question ${f 4}$

Incorrect

Mark 0.00 out of 1.00

Which of the following statements is True about the Bellman-Ford algorithm?

Select one or more:

- ☑ It always find if a negative weighted cycle is reachable from the source ✔
- ✓ None of the others X
- ☐ It always finds any cycle in the graph
- $\ \square$ It always finds if a negative weighted cycle exists

Your answer is incorrect.

The correct answer is: It always find if a negative weighted cycle is reachable from the source

	~ · ~		
Theory	()1117 7 .	Attempt	review

Question 5	
ncorrect Mark 0.00 out of 1.00	
Which of the following statements is	is True about the Floyd-Warshall algorithm?
Select one or more:	
$\ \square$ The runtime is $O(VE)$	
$\ oxdot$ The runtime is $O(V^3)$	
✓	
☑ The algorithm can only detect n	negative weight cycles reachable from the source 🗶
$\ \square$ The algorithm can not detect ne	egative weight cycles
Your answer is incorrect.	
The correct answer is: The runtime is	s $O(V^3)$
6	
Question 6	
Mark 0.00 out of 2.00	
What is the output (the maximum va	value that can be picked) of the Knapsack problem on the following input?
V 10 10 21 15	
W 5 3 7 6	
	items are given in the ${\bf V}$ and ${\bf W}$ arrays above (item i has value $V[i]$ and weight $W[i]$). Total capacity allowed to pick fractions of the above items.
Select one or more:	
□ 48	
□ 24	
✓ None of the others ★	
☑ 32 X	
□ 36	
Your answer is incorrect.	
The correct answer is: 36	

3 of 7

	~ · ~		
Theory	()11177	: Attemp	t review

Question 7	
Correct	
Mark 1.00 out of 1.00	
	nd B = "uvuwvwu". If the length of the longest common subsequence (not necessarily contiguous) e number of such longest common subsequences between P and Q is b, then the value of a + 10b is
Select one or more:	
□ 20	
☑ 34 ✔	
□ 42	
□ 28	
Your answer is correct.	
The correct answer is: 34	
Question 8	
Correct	
Mark 1.00 out of 1.00	
Which of following carrivor	be the Huffman encoding of three characters x,y, and z in a text file containing only those characters?
Select one or more: ☑ 0, 1, 00 ✓	be the Humilan encoding of three characters x,y, and 2 in a text me containing only those characters:
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔	be the Humilan encoding of three characters x,y, and 2 in a text me containing only those characters:
Select one or more: ✓ 0, 1, 00 ✓ ✓ 10, 01, 00 ✓ ☐ 0,10,11	be the Humilan encoding of three characters x,y, and 2 in a text me containing only those characters:
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔	be the Humilan encoding of three characters x,y, and 2 in a text me containing only those characters:
Select one or more: ✓ 0, 1, 00 ✓ ✓ 10, 01, 00 ✓ ☐ 0,10,11 ✓ 00,01,11 ✓	be the Humilian encoding of three characters x,y, and 2 in a text me containing only those characters:
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct.	
Select one or more: ✓ 0, 1, 00 ✓ ✓ 10, 01, 00 ✓ ✓ 0,10,11 ✓ 00,01,11 ✓	
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1	
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☑ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1	
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1	
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00	
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00	, 00, 10, 01, 00, 00,01,11
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☑ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible length	, 00, 10, 01, 00, 00,01,11
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible lengths are: Select one or more:	, 00, 10, 01, 00, 00,01,11
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible length of the correct of the	, 00, 10, 01, 00, 00,01,11
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible length of the others □ n/2 □ None of the others	, 00, 10, 01, 00, 00,01,11
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible length of the others ☐ n/2 ☐ None of the others ☑ n-1 ✔	, 00, 10, 01, 00, 00,01,11
Select one or more: ☑ 0, 1, 00 ✔ ☑ 10, 01, 00 ✔ ☐ 0,10,11 ☑ 00,01,11 ✔ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible length of the others ☑ n/2 ☐ None of the others ☑ n-1 ✔ ☐ n ☐ n(n-1)/2	, 00, 10, 01, 00, 00,01,11
Select one or more: ✓ 0, 1, 00 ✓ ✓ 10, 01, 00 ✓ O,10,11 ✓ 00,01,11 ✓ Your answer is correct. The correct answers are: 0, 1 Question 9 Correct Mark 1.00 out of 1.00 What is longest possible length of the others ✓ n-1 ✓ □ n	, 00, 10, 01, 00, 00,01,11

TD1	\sim .	^	.	
Theory	()1117.	2:	Attempt	review

In the coin change problem, we are interested in using the fewest number of coins to make change for a given amount. Suppose a country has the following coin denominations: 1, 4, 7, 13, 28, 52, 91, 365. A greedy algorithm repeatedly picks the coin of the largest denomination that does not exceed the target money. For e.g to make change for the amount 125, we use coins, 91, then 28, then 4 then two coins of 1. Which of the following statements are TRUE? Select one or more: None of the others This greedy algorithm always picks the optimum number of coins for this denomination of coins. This greedy algorithm never picks the optimum number of coins for this denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. The correct answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the		
denomination that does not exceed the target money. For e.g to make change for the amount 125, we use coins, 91, then 28, then 4 then two coins of 1. Which of the following statements are TRUE? Select one or more: None of the others This greedy algorithm always picks the optimum number of coins for this denomination of coins. This greedy algorithm never picks the optimum number of coins for this denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. The correct of	Mark 0.00 out of 2.00	
country has the following coin denominations: 1, 4, 7, 13, 28, 52, 91, 365. A greedy algorithm repeatedly picks the coin of the largest denomination that does not exceed the target money. For e.g to make change for the amount 125, we use coins, 91, then 28, then 4 then two coins of 1. Which of the following statements are TRUE? Select one or more: None of the others This greedy algorithm always picks the optimum number of coins for this denomination of coins. This greedy algorithm never picks the optimum number of coins for this denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. This greedy algorithm will always pick the optimum number of coins for any denomination of coins. Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins. We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight cycles. Any undirected graphs with no negative weight cycles. Any undirected graphs.		
Select one or more: None of the others This greedy algorithm always picks the optimum number of coins for this denomination of coins. This greedy algorithm never picks the optimum number of coins for this denomination of coins. Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins for any denomination of coins Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight edges. Any undirected graphs with no negative weight cycles. X Any undirected graph.	country has the followers denomination that de	wing coin denominations: 1, 4, 7, 13, 28, 52, 91, 365. A greedy algorithm repeatedly picks the coin of the largest
None of the others This greedy algorithm always picks the optimum number of coins for this denomination of coins. ★ This greedy algorithm never picks the optimum number of coins for this denomination of coins. Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins This greedy algorithm will always pick the optimum number of coins This greedy algorithm will always pick the optimum number of coins This greedy algorithm This greedy a	Which of the following	ng statements are TRUE?
□ This greedy algorithm always picks the optimum number of coins for this denomination of coins. ★ □ This greedy algorithm never picks the optimum number of coins for this denomination of coins. □ Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins ★ Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins Occurrent We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. ★ Any undirected graphs with no negative weight cycles. ★ Any undirected graphs.	Select one or more:	
This greedy algorithm never picks the optimum number of coins for this denomination of coins. Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins for any denomination of coins Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is algorithm to directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. ★ Any undirected graph.	☐ None of the other	ers
Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins This greedy algorithm will always pick the optimum number of coins for any denomination of coins Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins Ouestion 11 The correct wark 0.00 out of 1.00 We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. ★ Any undirected graphs.	☑ This greedy algo	rithm always picks the optimum number of coins for this denomination of coins. $f x$
This greedy algorithm will always pick the optimum number of coins for any denomination of coins Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins Duestion 11 Roorrect Wark 0.00 out of 1.00 We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. Any undirected graph. Your answer is incorrect.	☐ This greedy algo	rithm never picks the optimum number of coins for this denomination of coins.
Your answer is incorrect. The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins Question 11 Incorrect Mark 0.00 out of 1.00 We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. * Any undirected graphs.	☐ Only a Dynamic	programming algorithm can always pick the optimum number of coins for any denomination of coins
The correct answer is: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination of coins Question 11 Incorrect Mark 0.00 out of 1.00 We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. X Any undirected graphs.	☑ This greedy algo	rithm will always pick the optimum number of coins for any denomination of coins 🗶
Ouestion 11 Couestion 11 Couestion 11 We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: □ Undirected graphs with no negative weight edges. □ No undirected graphs with no negative weight cycles. ★ □ Any undirected graph.	Your answer is incorr	ect.
Duestion 11 Incorrect Mark 0.00 out of 1.00 We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. Any undirected graph.	The correct answer is	: Only a Dynamic programming algorithm can always pick the optimum number of coins for any denomination
We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. * Any undirected graph.	of coins	
We have seen how the Bellman-Ford algorithm can be used to solve the SSSP problem on directed graphs. We can try applying the algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graphs with no negative weight cycles. Undirected graphs with no negative weight cycles. Your answer is incorrect.		
algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graph Undirected graphs with no negative weight cycles. Any undirected graph.		
algorithm on undirected graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is replaced with two directed edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the weight of the original undirected edge. Using the above transformation, the Bellman-Ford algorithm is guaranteed to work correctly on: Select one or more: Undirected graphs with no negative weight edges. No undirected graph Undirected graphs with no negative weight cycles. Any undirected graph.		
Select one or more: ☐ Undirected graphs with no negative weight edges. ☐ No undirected graph ☑ Undirected graphs with no negative weight cycles. ★ ☐ Any undirected graph. Your answer is incorrect.		
 Undirected graphs with no negative weight edges. No undirected graph Undirected graphs with no negative weight cycles. ★ Any undirected graph. Your answer is incorrect.	algorithm on undirect replaced with two direct weight of the original	tted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the I undirected edge.
 □ No undirected graph ☑ Undirected graphs with no negative weight cycles. ★ □ Any undirected graph. Your answer is incorrect.	algorithm on undirect replaced with two direct weight of the original	tted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the I undirected edge.
 ☑ Undirected graphs with no negative weight cycles. ★ ☐ Any undirected graph. Your answer is incorrect.	algorithm on undirect replaced with two direct weight of the origina Using the above trans	tted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the I undirected edge.
☐ Any undirected graph. Your answer is incorrect.	algorithm on undirectoreplaced with two directors weight of the original Using the above transcelections or more:	sted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. sformation, the Bellman-Ford algorithm is guaranteed to work correctly on:
Your answer is incorrect.	algorithm on undirectoreplaced with two directors weight of the original Using the above transcellect one or more: Undirected grap	sted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. I work correctly on:
	algorithm on undirected replaced with two directed with two directed grap. Using the above transcript of the original Using the above transcript. Undirected grap. No undirected grap.	ted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. sformation, the Bellman-Ford algorithm is guaranteed to work correctly on: this with no negative weight edges.
The correct answer is: Undirected graphs with no negative weight edges.	algorithm on undirected grap algorithm on undirected grap Using the above trans Select one or more: Undirected grap Undirected grap Undirected grap	Inted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. In specific the same as the land of the same as th
	algorithm on undirected greplaced with two directed grap Using the above trans Select one or more: Undirected grap No undirected grap Undirected grap Any undirected grap	ted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. I undirected edges. I undirected edges. I undirected edges. I undirected edges is the same as the lundirected edges is the same as the lundirected edges. I undirected edges. I undirected edges is the same as the lundirected edges is the same as the lundirected edges. I undirected edges. I undirected edges. I undirected edges is the same as the lundirected edges is the same as the lundirected edges. I undirected edges. I undirected edges. I undirected edges. I undirected edges is the same as the lundirected edges is the same as the lundirected edges. I undirected edges. I und
	algorithm on undirected grap No undirected grap Undirected grap Any undirected grap Your answer is incorr	Inted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. In the Bellman-Ford algorithm is guaranteed to work correctly on: The weight of these two directed edges is the same as the lundirected edge. In the Bellman-Ford algorithm is guaranteed to work correctly on: The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundire
	algorithm on undirected grap No undirected grap Undirected grap Any undirected grap Your answer is incorr	Inted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. In the Bellman-Ford algorithm is guaranteed to work correctly on: The weight of these two directed edges is the same as the lundirected edge. In the Bellman-Ford algorithm is guaranteed to work correctly on: The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundire
	algorithm on undirect replaced with two directed grap Using the above transpace Select one or more: Undirected grap No undirected grap Any undirected grap Any undirected grap Your answer is incorrected grap Your answer is incorrected grap	Inted graphs by transforming them into directed graphs as follows: every (undirected) edge (u,v) in the graph is rected edges, one from u to v and other from v to u. The weight of these two directed edges is the same as the l undirected edge. In the Bellman-Ford algorithm is guaranteed to work correctly on: The weight of these two directed edges is the same as the lundirected edge. In the Bellman-Ford algorithm is guaranteed to work correctly on: The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of these two directed edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundirected edges is the same as the lundirected edges. The weight of the same as the lundire

TT1	\sim .			
Theory	()1117 \(\)	2: Atte	empt	review

Question 12 Incorrect	
Mark 0.00 out of 1.00	
Which of the following	statements are TRUE?
Select one or more:	
□ None of the above	
☐ The runtime of Gre	eedy algorithms is always more than the runtime of Dynamic Programming algorithms
✓ A larger number o	f combinatorial optimisation problems can be solved by Dynamic programming than by the Greedy strategy
☑ The runtime of Gre	eedy algorithms is always lesser than the runtime of Dynamic Programming algorithms 🗙
Your answer is incorrec	rt.
The correct answer is: A	A larger number of combinatorial optimisation problems can be solved by Dynamic programming than by the
Question 13	
Not answered	
Marked out of 2.00	
Consider an array of in	tegers A[1n]. We are interested in solving the following problems:
	e largest sum of elements in a contiguous subarray A[ij]
	e largest product of elements in a contiguous subarray A[ij]
	5, 0, 12, -7, 3}, the answer to Problem 1 is 16 (A[14])and Problem 2 is 480 (A[02]).
Which of the following	
Select one or more:	
☐ Problem 1 can be	most efficiently solved using Backtracking
☐ Problem 1 can be	most efficiently solved using Divide and Conquer
☐ Problem 2 can be	most efficiently solved using Dynamic Programming
☐ Problem 2 can be	most efficiently solved using Divide and Conquer
Your answer is incorrec	rt.
	e: Problem 1 can be most efficiently solved using Divide and Conquer, Problem 2 can be most efficiently solved

	~ · ~		
Theory	()1117 ⁽)	 Attem: 	pt review

Question 14 Not answered	
farked out of 2.00	
	of finding the longest monotonically increasing sequence of numbers in an array A[0n-1]. One possible way to follows : (Let l_i denote the length of the longest monotonically increasing sequence that starts at index i in
initialize l_{n-1} to 1.	
for $i=n-2\ to\ 0$	
if then $l_i=1$	$+\ l_{i+1}$ else $l_i=1$
return $\max(l_0, l_1, \dots l_n)$	$_{-1})$
Which of the following	statements are TRUE?
Select one or more:	
☐ The condition in th	e blank space is "A[i-1] < A[i]"
☐ None of the others	
☐ This algorithm uses	s Dynamic Programming
☐ The condition in th	e blank space is "A[i] $<$ A[i+1]"
☐ This is a Backtracki	ng algorithm
The correct answers are	:: The condition in the blank space is "A[i] < A[i+1]", This algorithm uses Dynamic Programming
Question 15	e: The condition in the blank space is "A[i] < A[i+1]", This algorithm uses Dynamic Programming
Question 15 Not answered	e: The condition in the blank space is "A[i] < A[i+1]", This algorithm uses Dynamic Programming
Question 15	e: The condition in the blank space is "A[i] < A[i+1]", This algorithm uses Dynamic Programming
Question 15 Not answered Marked out of 1.00 Recall the Greedy algor	e: The condition in the blank space is "A[i] < A[i+1]", This algorithm uses Dynamic Programming ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the
Question 15 Not answered Marked out of 1.00 Recall the Greedy algor overlapping intervals a	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non-
Question 15 Not answered Marked out of 1.00 Recall the Greedy algor overlapping intervals at same problem? Select one or more:	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non-
Duestion 15 Not answered Marked out of 1.00 Recall the Greedy algor overlapping intervals as same problem? Select one or more:	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the
Recall the Greedy algor overlapping intervals at same problem? Select one or more: If no intervals conf Choose the interval	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse.
Recall the Greedy algor overlapping intervals at same problem? Select one or more: If no intervals conf Choose the interval	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse I that ends last, discard all its conflicting intervals and then recurse
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse I that ends last, discard all its conflicting intervals and then recurse
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others Choose the interval	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse I that ends last, discard all its conflicting intervals and then recurse
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others Choose the interval	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse I that ends last, discard all its conflicting intervals and then recurse. I that starts last, discard all its conflicting intervals and then recurse.
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others Choose the interval	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse I that ends last, discard all its conflicting intervals and then recurse. I that starts last, discard all its conflicting intervals and then recurse. It choose the interval that starts last, discard all its conflicting intervals and then recurse.
Recall the Greedy algor overlapping intervals as same problem? Select one or more: If no intervals conf Choose the interval None of the others Choose the interval	ithm for scheduling a set of intervals (with a given start and finish time) such that a maximum number of non- re scheduled on a single resource. Which of the following greedy algorithms would also correctly solve the lict, choose them all. Else, discard the interval with the longest duration and recurse. I that starts first, discard all its conflicting intervals and then recurse I that ends last, discard all its conflicting intervals and then recurse. I that starts last, discard all its conflicting intervals and then recurse. It choose the interval that starts last, discard all its conflicting intervals and then recurse.

7 of 7