

MODEL PAPER 2

Unofficial practice paper modelled on the AY21 (2024) and AY22 (2025) past papers

University of Colombo School of Computing BACHELOR OF SCIENCE IN COMPUTER SCIENCE

First Year Examination – Semester I

SCS1304 — Problem Solving Strategies and Computational Approaches

(Two (2) Hours)

Answer ALL questions

Number of Pages = 18

Number of Questions = 3

To be completed by the candidate							
Index Number:							

Important Instructions to Candidates:

- I. Students should answer in the medium of English language only using the space provided in this question paper.
- II. Note that questions appear on both sides of the paper. If a page or a part of this question paper is not printed, please inform the supervisor immediately.
- III. Write your index number CLEARLY on each and every page of this question paper.
- IV. This paper consists of three (03) questions in 18 pages (including the Cover Page). The first question consists of 20 MCQs.
- V. Answer ALL questions.
- VI. Programmable calculators and any electronic device capable of storing and retrieving text, including electronic dictionaries, smartwatches, and mobile phones, are **not allowed**.
- VII. Non-programmable calculators are also **not allowed**.
- VIII. Do not tear off any part of this answer book. Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- IX. **The first question carries 40 marks, and the second and third questions carry 30 marks each.**

To be completed by the examiners

1	
2	
3	
Total	

Index No.								
-----------	--	--	--	--	--	--	--	--

Q1.	i. Question 1 has 20 MCQs. ii. Each question carries 1 mark. iii. Each MCQ has 5 (five) choices with one correct answer. iv. Mark answers for MCQs on the MCQ answer sheet provided. (40 marks)
------------	--

1)	Given the computational effort to solve a task as $4n \log n + 3n + 50$, where n is the size of its input, what is its time complexity?
(a)	$O(n)$
(b)	$O(\log n)$
(c)	$O(n^2)$
(d)	$O(n \log n)$
(e)	$O(n^2 \log n)$

2)	What is the time complexity of the following evaluation? <pre> total = 0; for (int i = 1; i <= n; i *= 2) { for (int j = 1; j <= i; j++) { total++; } } </pre>
(a)	$O(n)$
(b)	$O(\log n)$
(c)	$O(n \log n)$
(d)	$O(n^2)$
(e)	$O((\log n)^2)$

3)	What is the outcome if a Greedy approach is used to determine the optimal set of coins (i.e., minimum number of coins) to make 48 Rupees, using coin denominations {1, 5, 10, 20}?
(a)	20, 20, 5, 3
(b)	20, 10, 10, 5, 1, 1, 1
(c)	20, 20, 5, 1, 1, 1
(d)	20, 20, 4, 4
(e)	10, 10, 10, 10, 5, 1, 1, 1

Index No.									
-----------	--	--	--	--	--	--	--	--	--

4)	Which of the following statements express the <i>Overlapping Sub-problems</i> property of a problem as used in Dynamic Programming? A. Fibonacci: $\text{fib}(n - 2)$ is computed both inside $\text{fib}(n)$ and inside $\text{fib}(n - 1)$. B. Shortest path: a shortest path from s to t through v contains a shortest path from s to v. C. LCS: the value $L(i - 1, j - 1)$ is needed when computing $L(i, j)$, $L(i - 1, j)$ and $L(i, j - 1)$. D. Merge Sort: the two halves are sorted independently and never share a sub-problem.	
	(a)	A only
	(b)	A and C only
	(c)	B and D only
	(d)	A, B and C only
	(e)	All of A, B, C and D

5)	What is the time complexity of the following evaluation? <pre> total = 0; for (int i = 0; i < n; i++) { for (int j = n; j > 0; j /= 2) { for (int k = 0; k < j; k++) { total++; } } } </pre>	
	(a)	$O(n \log n)$
	(b)	$O((\log n)^2)$
	(c)	$O(n^2 \log n)$
	(d)	$O(n^3)$
	(e)	$O(n^2)$

6)	Given an array $A = \{3, 8, 15, 21, 27, 34, 40, 46, 52, 60, 67, 73, 81, 88, 95\}$ and $\text{key} = 21$, what are the mid values (corresponding array elements) in the first and second levels of recursion respectively, in <i>binary search</i>? Assume $\text{mid} = \lfloor (\text{low} + \text{high})/2 \rfloor$.	
	(a)	46 and 21
	(b)	46 and 15
	(c)	40 and 21
	(d)	46 and 27
	(e)	52 and 21

Index No.									
-----------	--	--	--	--	--	--	--	--	--

7)	Consider the following two sequences. S1 = {A, G, G, T, A, B} S2 = {G, X, T, X, A, Y, B} What is the <i>Longest Common Subsequence (LCS)</i> of S1 and S2?	
	(a)	{G, A, B}
	(b)	{G, X, A, B}
	(c)	{A, G, T, B}
	(d)	{G, T, A, B}
	(e)	{G, T, X, B}
8)	Which of the following statements relating to <i>Backtracking</i> and <i>Branch and Bound</i> are correct? A. Backtracking typically explores the state space tree in a depth-first manner. B. Branch and bound uses bounds on the cost of a solution to prune nodes in optimisation problems. C. Branch and bound can only explore nodes in Last-In-First-Out (LIFO) order. D. Both techniques can discard branches that cannot lead to a (better) solution.	
	(a)	A and B only
	(b)	A, B and D only
	(c)	B and C only
	(d)	A, C and D only
	(e)	All of A, B, C and D
9)	Which of the following statement(s) is/are correct regarding <i>Dijkstra's</i> shortest path algorithm? A. It is a greedy algorithm. B. At each step it selects the unvisited vertex with the smallest tentative distance. C. It always gives correct shortest paths even when some edge weights are negative.	
	(a)	A only
	(b)	B and C only
	(c)	A and B only
	(d)	A and C only
	(e)	All of A, B and C
10)	Which of the following statements is correct regarding $f(n) = 2n^2 + 3n$? Note n is a natural number representing problem size.	
	(a)	$f(n) \in \Theta(n)$ because the $3n$ term is linear
	(b)	$f(n) \in \Theta(n^2)$ because $2n^2 \leq f(n) \leq 5n^2$ for all $n \geq 1$
	(c)	$f(n) \in \Theta(n^3)$ because $f(n) \leq n^3$ for all $n \geq 3$
	(d)	$f(n) \notin O(n^2)$ because the $3n$ term keeps growing
	(e)	$f(n) \in \Theta(n^2)$ because $f(n) \leq 2n^2$ for all $n \geq 1$

Index No.									
-----------	--	--	--	--	--	--	--	--	--

11)	Which of the following statement(s) is/are correct regarding the <i>Divide and Conquer</i> approach? A. The problem is divided into smaller sub-problems of the same type. B. The solutions of the sub-problems are combined to obtain the solution of the original problem. C. The sub-problems typically overlap heavily, so their results must always be stored.	
	(a)	A and B only
	(b)	A only
	(c)	B and C only
	(d)	A and C only
	(e)	All of A, B and C

12)	Using the following partition function (pivot = last element), what is the state of the array [7, 2, 9, 4, 3, 8, 5] after the first call partition(a, 0, 6) of <i>Quick Sort</i>? <pre> int partition(int a[], int low, int high) { int pivot = a[high], i = low - 1; for (int j = low; j < high; j++) if (a[j] <= pivot) { i++; swap(&a[i], &a[j]); } swap(&a[i + 1], &a[high]); return i + 1; } </pre>	
	(a)	[2, 3, 4, 5, 7, 8, 9]
	(b)	[2, 4, 3, 5, 7, 9, 8]
	(c)	[4, 2, 3, 5, 9, 8, 7]
	(d)	[2, 4, 3, 7, 9, 8, 5]
	(e)	[2, 4, 3, 5, 9, 8, 7]

13)	What is the time complexity of the following evaluation? <pre> total = 0; for (int i = 1; i <= n; i++) { for (int j = 1; j <= n; j += i) { total++; } } </pre>	
	(a)	$O(n^2)$
	(b)	$O(n)$
	(c)	$O(n \log n)$
	(d)	$O(\log n)$
	(e)	$O(n\sqrt{n})$

Index No.									
-----------	--	--	--	--	--	--	--	--	--

14)	Consider the following function: <pre>void show(int n) { if (n <= 0) return; show(n - 1); printf("%d ", n); }</pre> Which of the following is correct regarding <code>show(3)</code>?	
	(a)	It is tail recursion and prints 3 2 1
	(b)	It is head recursion and prints 1 2 3
	(c)	It is head recursion and prints 3 2 1
	(d)	It is tail recursion and prints 1 2 3
	(e)	It is nested recursion and prints 1 2 3

15)	Consider the following function: <pre>int power(int x, int n) { if (n == 0) return 1; int h = power(x, n / 2); if (n % 2 == 0) return h * h; else return x * h * h; }</pre> Which of the following is correct regarding the above? A) Its time complexity is $O(\log n)$. B) It makes two recursive calls per invocation, so it is tree recursion. C) Calling <code>power(2, 5)</code> results in a total of 4 invocations of <code>power</code> (including the first call). D) Its space complexity is $O(\log n)$ due to the call stack.	
	(a)	A and B only
	(b)	B, C and D only
	(c)	A and D only
	(d)	A, C and D only
	(e)	All of A, B, C and D

16)	A thief has a knapsack of capacity 50 kg and can take fractions of items. The items have (weight, value) pairs (10 kg, Rs.60), (20 kg, Rs. 100) and (30 kg, Rs. 120). What is the maximum value obtained using the <i>Greedy</i> approach based on value/weight ratio?	
	(a)	Rs. 240
	(b)	Rs. 220
	(c)	Rs. 160
	(d)	Rs. 280
	(e)	Rs. 180

Index No.									
-----------	--	--	--	--	--	--	--	--	--

17)	Consider the following statements regarding <i>Insertion Sort</i>: A) It is a stable sorting algorithm. B) It sorts in-place. C) Its worst-case time complexity is $O(n^2)$. D) Its best-case time complexity is $O(n \log n)$. Which of the above is/are correct?	
	(a)	A only
	(b)	A and B only
	(c)	A, B and C only
	(d)	B, C and D only
	(e)	All of A, B, C and D

18)	Consider the Insertion Sort algorithm below: <pre> void insertionSort(int arr[], int n) { for (int i = 1; i < n; i++) { int key = arr[i], j = i - 1; while (j >= 0 && arr[j] > key) { arr[j + 1] = arr[j]; j--; } arr[j + 1] = key; } } </pre> What is the state of the array [6, 3, 8, 2, 5] after the outer loop has completed the iteration $i = 3$?	
	(a)	[3, 6, 8, 2, 5]
	(b)	[2, 3, 6, 8, 5]
	(c)	[2, 3, 5, 6, 8]
	(d)	[3, 6, 2, 8, 5]
	(e)	[2, 6, 3, 8, 5]

19)	For an algorithm with time complexity $O(2^n)$, if the input size increases from n to $n + 1$, the running time approximately ...	
	(a)	increases by 1.
	(b)	squares.
	(c)	increases by a factor of n .
	(d)	remains constant.
	(e)	doubles.

20)	How many distinct solutions exist for the <i>4-Queens</i> problem on a 4×4 chessboard?	
	(a)	1
	(b)	2
	(c)	4
	(d)	8
	(e)	0

Index No.								
-----------	--	--	--	--	--	--	--	--

Q2.	A.	A university needs to prepare the timetable for its annual inter-faculty sports meet (events, venues, officials and time slots). (i) List the four pillars of <i>Computational Thinking</i>. (ii) Briefly explain how <i>Decomposition</i> and <i>Pattern Recognition</i> can be applied to the above problem, giving an example for each. (06 marks)
-----	----	---

ANSWER BOX

(i)

.....

.....

(ii) Decomposition:

.....

.....

.....

Example:

.....

.....

Pattern Recognition:

.....

.....

.....

Example:

.....

.....

Index No.									
-----------	--	--	--	--	--	--	--	--	--

Q2.

B.

A delivery rider has a bag that can carry at most **20 kg**. The following goods are available, and any good can be **divided** (a fraction of it may be taken).

Item	I1	I2	I3	I4	I5	I6
Weight (kg)	4	6	5	8	3	10
Value (Rs. '000)	20	30	40	32	21	40

(i) Using a *Greedy Algorithm*, find the items (and fractions) to be selected to maximise the total value. Show your steps and state any assumptions made.

(ii) If items **cannot** be divided (0/1 Knapsack), does the same greedy strategy always give the optimal value for the above data? Justify your answer.

(06 marks)

ANSWER BOX

Selected items =

Maximum value =

(ii)

.....

.....

Index No.									
-----------	--	--	--	--	--	--	--	--	--

Q2.	C.	The <i>Divide-and-conquer</i> approach can be used in the <i>Merge Sort</i> algorithm. Draw and illustrate how a human would sort the following array using the <u>Merge Sort</u> algorithm. State any assumptions made. <table><tr><td>15</td><td>4</td><td>23</td><td>8</td><td>16</td><td>42</td><td>7</td></tr></table>	15	4	23	8	16	42	7
15	4	23	8	16	42	7			
		(06 marks)							

ANSWER BOX

Assumptions (if any):

.....

.....

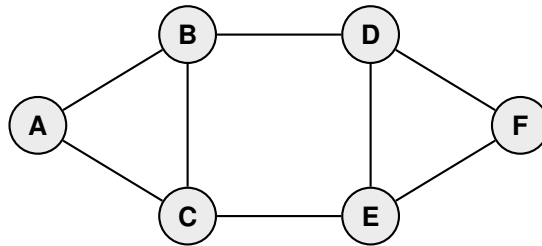
Illustration:

Index No.								
-----------	--	--	--	--	--	--	--	--

Q2.

D.

Consider the following undirected graph G with 6 vertices:



Using the *Backtracking method*, show step by step how to find *Hamiltonian Paths* of G starting from vertex **A**. Draw the partial search tree as the algorithm explores paths. Show **one** successful *Hamiltonian Path* and **one** dead end.

(06 marks)

ANSWER BOX

Path found:

Dead end:

Index No.								
-----------	--	--	--	--	--	--	--	--

Q2.	E.	Given the set $S = \{3, 5, 6, 7\}$ (sorted in ascending order) and a target sum $T = 15$, it is required to find all subsets of S whose elements add up exactly to T using <i>Backtracking</i>. (i) Draw the <i>state-space tree</i> (include / exclude each element in order), clearly marking the nodes that are pruned. (ii) State the two bounding conditions used to prune the tree. (iii) List all the subset(s) found. (06 marks)
-----	----	--

ANSWER BOX

(ii)

.....

.....

(iii)

.....

Index No.									
-----------	--	--	--	--	--	--	--	--	--

Q3.	A.	Consider the following <i>Bottom-Up Dynamic Programming (tabulation)</i> algorithm for finding the length of the <i>Longest Common Subsequence (LCS)</i> of two strings X (length m) and Y (length n): <pre> int lcsLength(char X[], char Y[], int m, int n) { int L[m + 1][n + 1]; for (int i = 0; i <= m; i++) { for (int j = 0; j <= n; j++) { if (i == 0 j == 0) L[i][j] =A.....; else if (X[i - 1] ==B.....) L[i][j] =C..... + 1; else L[i][j] = max(.....D.....,E.....); } } return L[m][n]; } </pre> (i) What are the appropriate terms for the spaces labelled A to E? (ii) What is the time and space complexity of the above algorithm? State reasons for your answers. (iii) Fill in the table L for X = "BDCB" and Y = "BACDB", and write down one LCS. (07 marks)
------------	-----------	--

ANSWER BOX

(i)

A:

B:

C:

D:

E:

Index No.								
-----------	--	--	--	--	--	--	--	--

(ii)

Time Complexity:

Space Complexity:

(iii)

X \ Y	∅	B	A	C	D	B
∅						
B						
D						
C						
B						

One LCS:

Q3.

B.

Consider the following adjacency matrix to be used in a *Branch and Bound Least Cost* approach in the *Travelling Salesman Problem (TSP)* algorithm. Note that A to D are nodes of the undirected graph.

	A	B	C	D
A	∞	5	9	6
B	5	∞	4	8
C	9	4	∞	3
D	6	8	3	∞

Starting from node A, calculate the relevant cost matrices and find out the cost of nodes A and B in the state space tree. Fill in the spaces given as appropriate and state any assumptions made.

(06 marks)

ANSWER BOX

Calculate the cost of root node A by filling in the spaces given below.

Row Reduced Matrix

	A	B	C	D
A				
B				
C				
D				

Column Reduced Matrix

	A	B	C	D
A				
B				
C				
D				

The Cost of node A =

Calculate the cost of Node B by filling in the spaces given below.

To add the edge from A to B in the state space tree, write down the resultant cost matrices.

Row Reduced Matrix

	A	B	C	D
A				
B				
C				
D				

Column Reduced Matrix

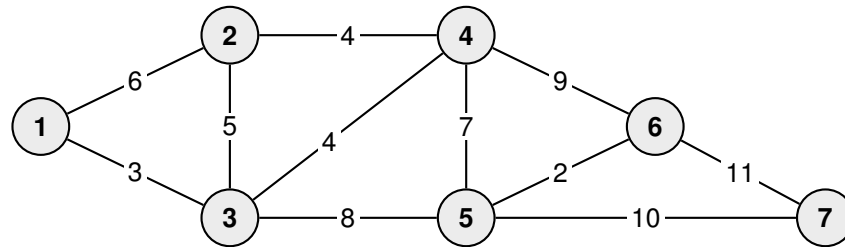
	A	B	C	D
A				
B				
C				
D				

The Cost of node B is =

Q3.

C.

Prim's algorithm is a *Greedy* approach that generates a Minimum Spanning Tree (MST) for a weighted undirected graph. Show steps and draw the MST(s) and the total weight(s) of edges for each MST, for the following graph using *Prim's algorithm*. State any assumptions made. What is the minimum possible total weight of the MST?



(05 marks)

ANSWER BOX

The minimum possible total weight of MST:

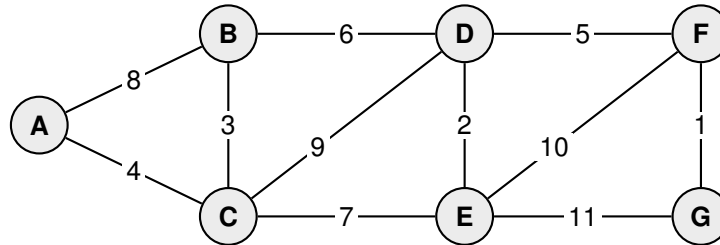
.....

Index No.								
-----------	--	--	--	--	--	--	--	--

Q3

D.

Kruskal's algorithm is another *Greedy* approach for finding the Minimum Spanning Tree (MST) of an undirected graph. Show steps and draw the MST and the minimum possible total weight of the following graph using *Kruskal's algorithm*. State any assumptions made.



(05 marks)

ANSWER BOX

The minimum possible total weight of MST:

.....

