

MODEL PAPER 1

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 = 17

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 17 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. Select the most correct answer. v. Mark answers on the MCQ answer sheet provided separately. (40 marks)
------------	--

1)	Which of the following shows the correct order of growth of time complexity?
(a)	$O(1) < O(\log n) < O(n) < O(\sqrt{n}) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$
(b)	$O(1) < O(\sqrt{n}) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n!) < O(2^n)$
(c)	$O(1) < O(\log n) < O(\sqrt{n}) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$
(d)	$O(\log n) < O(1) < O(\sqrt{n}) < O(n) < O(n^2) < O(n \log n) < O(2^n) < O(n!)$
(e)	$O(1) < O(\log n) < O(\sqrt{n}) < O(n) < O(n \log n) < O(2^n) < O(n^2) < O(n!)$

2)	If the input size of an algorithm with time complexity $O(n^2)$ is tripled , the running time increases by a factor of approximately ...
(a)	Three
(b)	Six
(c)	Twenty-seven
(d)	Nine
(e)	A negligible amount

3)	A problem can be solved efficiently using <i>Dynamic Programming</i> when it has ...
(a)	the greedy choice property and optimal substructure.
(b)	overlapping sub-problems and optimal substructure.
(c)	overlapping sub-problems and the greedy choice property.
(d)	independent sub-problems and a pruning function.
(e)	a sorted input and optimal substructure.

4)	For which of the following problems does a <i>Greedy Algorithm</i> not guarantee an optimal solution?
(a)	Fractional Knapsack
(b)	Activity Selection
(c)	Minimum Spanning Tree using Kruskal's algorithm
(d)	Minimum Spanning Tree using Prim's algorithm
(e)	0/1 Knapsack

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

5)	What is the time complexity of finding the <i>Longest Common Subsequence</i> of two sequences of lengths m and n using <i>Dynamic Programming</i> (tabulation)?	
	(a)	$O(mn)$
	(b)	$O(m + n)$
	(c)	$O(2^{m+n})$
	(d)	$O(m \log n)$
	(e)	$O(\max(m, n))$

6)	Consider the following statements regarding <i>Memoization</i> (top-down) and <i>Tabulation</i> (bottom-up) in <i>Dynamic Programming</i>. A. Memoization only computes the sub-problems that are actually required to solve the original problem. B. Tabulation is usually implemented iteratively, filling a table in a fixed order from the smallest sub-problems. C. Memoization can never cause a stack overflow because the results of sub-problems are stored. Which of the above statement(s) is/are correct?	
	(a)	A only
	(b)	A and B only
	(c)	B and C only
	(d)	A and C only
	(e)	All A, B and C

7)	Consider the following two sequences: S1 = {A, B, C, B, D, A, B} S2 = {B, D, C, A, B, A} Which of the following is a <i>Longest Common Subsequence (LCS)</i> of S1 and S2?	
	(a)	{B, C, A}
	(b)	{A, B, B, A}
	(c)	{B, C, B, A}
	(d)	{B, D, B, A}
	(e)	{B, C, D, A}

8)	What is the maximum number of comparisons needed to find a key in a sorted array of 1000 elements using <i>Binary Search</i> ?	
	(a)	9
	(b)	10
	(c)	11
	(d)	500
	(e)	1000

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

9)	Consider the following statements regarding the <i>Divide and Conquer</i> approach.	
	A. The recurrence $T(n) = 2T(n/2) + O(n)$ of <i>Merge Sort</i> gives a time complexity of $O(n \log n)$.	
	B. The recurrence $T(n) = T(n/2) + O(1)$ of <i>Binary Search</i> gives a time complexity of $O(\log n)$.	
	C. <i>Strassen's algorithm</i> makes eight recursive multiplications of $n/2 \times n/2$ matrices.	
	Which of the above statement(s) is/are correct ?	
	(a)	A only
10)	(b)	B only
	(c)	A and B only
	(d)	B and C only
	(e)	All A, B and C
	Consider the following pseudocode that recursively calculates the sum of the first n elements of an array A, with missing terms P and Q :	
	<pre> Integer sum(A, n) { if n == 0: return 0 returnP.... +Q.... } </pre>	
	Which of the following correctly replace the labels P and Q?	
	(a)	$P \rightarrow A[n], Q \rightarrow \text{sum}(A, n)$
	(b)	$P \rightarrow A[0], Q \rightarrow \text{sum}(A, n)$
	(c)	$P \rightarrow n, Q \rightarrow \text{sum}(A, n - 1)$
	(d)	$P \rightarrow A[n - 1], Q \rightarrow \text{sum}(A, n - 1)$
	(e)	$P \rightarrow A[n - 1], Q \rightarrow \text{sum}(A, n - 2)$
11)	Assume that you have an infinite number of coins of denominations $\{1, 4, 6\}$. A <i>Greedy Algorithm</i> always picks the largest coin that does not exceed the remaining amount. To make an amount of 8, how many coins does the greedy algorithm use, and what is the minimum number of coins possible?	
	(a)	2 and 2
	(b)	3 and 2
	(c)	3 and 3
	(d)	4 and 2
	(e)	2 and 3

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

12)	Consider the following statements regarding <i>Merge Sort</i> on arrays. A. Its time complexity is $O(n \log n)$ in the best, average and worst cases. B. The standard implementation is stable. C. The standard implementation is in-place, requiring only $O(1)$ extra memory. Which of the above statement(s) is/are correct?	
	(a)	A and B only
	(b)	A only
	(c)	B and C only
	(d)	A and C only
	(e)	All A, B and C

13)	What is the time complexity of the following? <pre> total = 0; for (int i = 0; i < n; i++) { for (int j = i; j < n; j++) { for (int k = 1; k < n; k *= 3) { total++; } } } </pre>	
	(a)	$O(n^2)$
	(b)	$O(n^3)$
	(c)	$O(n \log n)$
	(d)	$O(n^2 \log n)$
	(e)	$O(n(\log n)^2)$

14)	Consider the following two functions: <pre> void funA(int n) { if (n > 0) { printf("%d ", n); funB(n - 1); } } void funB(int n) { if (n > 1) { printf("%d ", n); funA(n / 2); } } </pre> Which type of recursion is shown above?	
	(a)	Tail recursion
	(b)	Indirect recursion
	(c)	Nested recursion
	(d)	Tree recursion
	(e)	Head recursion

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

15)	The following four steps, P, Q, R and S, outline <i>Prim's algorithm</i>. Note that they are not in the correct order. P - Among all edges joining a visited vertex to an unvisited vertex, select the edge with the minimum weight. Q - Choose a starting vertex and mark it as visited. R - Repeat until all the vertices are visited. S - Add the selected edge to the MST and mark its unvisited end vertex as visited. Which of the following is the correct order of steps?								
	(a)	P, Q, R, S							
	(b)	Q, S, P, R							
	(c)	Q, P, S, R							
	(d)	S, P, Q, R							
	(e)	Q, P, R, S							

16)	What is the state of the array [5, 1, 4, 2, 8] after the first pass of <i>Bubble Sort</i> (ascending order)?								
	(a)	[1, 4, 2, 5, 8]							
	(b)	[1, 5, 4, 2, 8]							
	(c)	[1, 2, 4, 5, 8]							
	(d)	[1, 4, 5, 2, 8]							
	(e)	[5, 1, 4, 2, 8]							

17)	What is the time complexity of a function $f(n) = 5n^3 + 2^n + 100n \log n$?								
	(a)	$O(n^3)$							
	(b)	$O(n \log n)$							
	(c)	$O(n^3 2^n)$							
	(d)	$O(n^4)$							
	(e)	$O(2^n)$							

18)	What is the best-case time complexity of <i>Insertion Sort</i>, and when does it occur?								
	(a)	$O(n \log n)$, when the array is already sorted							
	(b)	$O(n^2)$, when the array is sorted in reverse order							
	(c)	$O(\log n)$, when the array is already sorted							
	(d)	$O(n)$, when the array is already sorted							
	(e)	$O(n)$, when the array is sorted in reverse order							

19)	What does it mean when we say that $f(n) = \Theta(g(n))$?								
	(a)	$f(n)$ is bounded only from above by a constant multiple of $g(n)$.							
	(b)	For large n , $f(n)$ is bounded both above and below by constant multiples of $g(n)$.							
	(c)	$f(n)$ is bounded only from below by a constant multiple of $g(n)$.							
	(d)	$f(n) = g(n)$ for all values of n .							
	(e)	$f(n)$ grows strictly slower than $g(n)$.							

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

20)	In <i>Least Cost (LC) Branch and Bound</i> , the next live node to be expanded (E-node) is selected ...	
	(a)	in the order in which the nodes were generated.
	(b)	as the most recently generated node.
	(c)	in random order.
	(d)	as the live node with the least estimated cost.
	(e)	as the deepest node in the state space tree.

Q2.	A.	Briefly explain how <i>Abstraction</i> and <i>Algorithmic Thinking (Algorithm Design)</i> are used in <i>Computational Thinking</i> . Provide an example for each in the context of designing a mobile app that plans bus journeys within Colombo.
		(06 marks)

ANSWER BOX

Abstraction:

.....

Example:

.....

Algorithmic Thinking:

.....

Example:

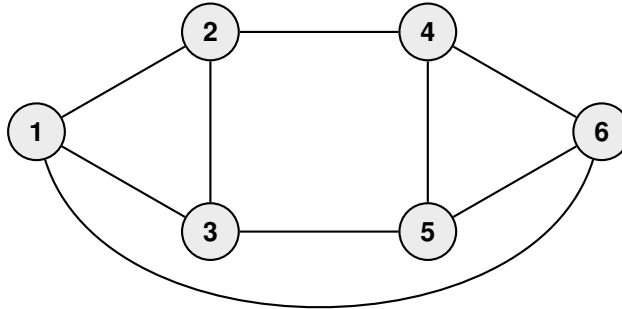
.....

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

Q2.

B.

A **Hamiltonian Cycle** is a cycle in a graph that visits every vertex exactly once and returns to the starting vertex. The *backtracking* approach uses a *state-space tree* to check whether a Hamiltonian cycle exists. Draw a State Space Tree showing backtracking and find all the Hamiltonian Cycle(s) of the following graph starting at **vertex 1**. State any assumptions made.



(07 marks)

ANSWER BOX

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

Q2.	C.	The <i>divide-and-conquer</i> approach is used in the <i>Quick Sort</i> algorithm. Draw and illustrate how a human would sort the following array using <i>Quick Sort</i>. Assume that the <u>pivot</u> is always chosen as the <u>last item</u>. <table><tr><td>21</td><td>6</td><td>14</td><td>9</td><td>25</td><td>3</td><td>17</td><td>12</td></tr></table> (07 marks)	21	6	14	9	25	3	17	12
21	6	14	9	25	3	17	12			

ANSWER BOX

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

Q2.	D.	The binomial coefficient $C(n, k)$ can be defined recursively as $C(n, k) = C(n - 1, k - 1) + C(n - 1, k)$, with $C(n, 0) = C(n, n) = 1$. (i) Write down a recursive pseudocode to calculate $C(n, k)$. (ii) Draw the recursion tree for calculating $C(4, 2)$. (iii) How many times is $C(2, 1)$ computed in part (ii)? Briefly explain how <i>memoization</i> would improve this algorithm. (10 marks)
------------	-----------	---

ANSWER BOX

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

Q3.	A.	(i) Study the following code for the <i>Merge Sort</i> algorithm and write the relevant terms for labels P, Q, R, S and T in the given spaces. (ii) What are the <i>Worst, Average</i> and <i>Best Case</i> time complexities, and the auxiliary space complexity, of Merge Sort (you are not required to derive them)? State any assumptions made. <pre> void merge(int arr[], int l, int m, int r) { int n1 = m - l + 1, n2 = r - m; int L[n1], R[n2]; for (int i = 0; i < n1; i++) L[i] = arr[l + i]; for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j]; int i = 0, j = 0, k = l; while (i < n1 && j < n2) { if (L[i] <=S....) arr[k++] = L[i++]; else arr[k++] = R[j++]; } while (i < n1) arr[k++] =T....; while (j < n2) arr[k++] = R[j++]; } void mergeSort(int arr[], int l, int r) { if (l < r) { int m =P....; mergeSort(arr, l,Q....); mergeSort(arr,R...., r); merge(arr, l, m, r); } } </pre>
		(08 marks)

ANSWER BOX

(i)

P:

Q:

R:

S:

T:

(ii)

Worst Case:

Best Case:

Average Case:

Space Complexity:

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

Q3.

B.

Consider the following adjacency matrix to be used in a *Branch and Bound Least Cost* approach for the *Travelling Salesman Problem (TSP)*. Note that N_0 to N_4 are nodes of the graph.

	N_0	N_1	N_2	N_3	N_4
N_0	INF	14	9	21	12
N_1	10	INF	17	6	3
N_2	8	11	INF	5	13
N_3	16	4	20	INF	7
N_4	9	15	2	18	INF

Starting from N_0 , calculate the relevant cost matrices and find the cost of the root node N_0 and the node N_2 (edge $N_0 \rightarrow N_2$) in the *state space tree*. Fill in the spaces given as appropriate and state any assumptions made.

(07 marks)

ANSWER BOX

Calculate the cost of root node N_0 by filling the spaces given below.

Row Reduced Matrix

	N_0	N_1	N_2	N_3	N_4
N_0					
N_1					
N_2					
N_3					
N_4					

Column Reduced Matrix

	N_0	N_1	N_2	N_3	N_4
N_0					
N_1					
N_2					
N_3					
N_4					

The cost of node N_0 is =

.....

Calculate the cost of node N_2 by filling the spaces given below.

Now add the edge from N_0 to N_2 in the *state space tree* and write down the resultant cost matrix below.

	N_0	N_1	N_2	N_3	N_4
N_0					
N_1					
N_2					
N_3					
N_4					

Row Reduced Matrix

	N_0	N_1	N_2	N_3	N_4
N_0					
N_1					
N_2					
N_3					
N_4					

Index No.

Column Reduced Matrix

	N_0	N_1	N_2	N_3	N_4
N_0					
N_1					
N_2					
N_3					
N_4					

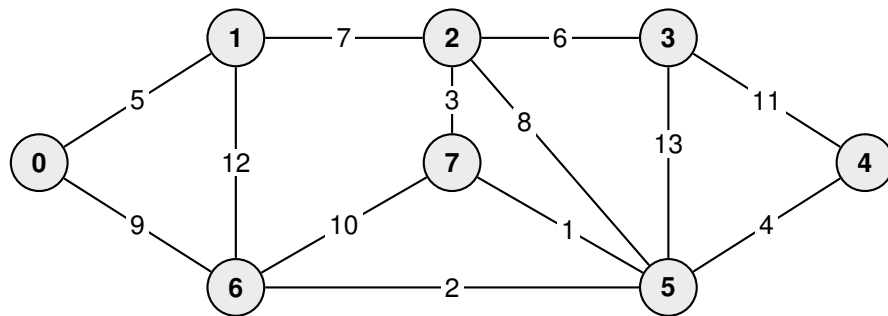
The cost of node N_2 is =

.....

Q3.

C.

Prim's algorithm is a *Greedy Algorithm* that finds 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 the following graph using **Prim's algorithm**, starting from vertex 0. State any assumptions made.



(08 marks)

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

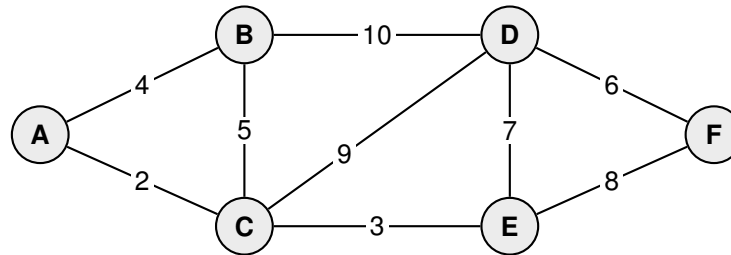
ANSWER BOX

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

Q3

D.

Kruskal's algorithm is a greedy algorithm for finding the *Minimum Spanning Tree (MST)* of a graph. Show steps and draw the MST and the relevant cost of the following graph using ***Kruskal's algorithm***. State any assumptions made.



(07 marks)

ANSWER BOX

MST

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

The Cost of MST:

.....

*****End of the Paper*****