Register Number

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Department of Computer Science and Engineering

Continuous Assessment Test – I

Question Paper

<table>
<tr><td>Degree & Branch</td><td colspan="3">B.E CSE</td><td>Semester</td><td>IV</td></tr>
<tr><td>Subject Code & Name</td><td colspan="3">UCS1403 – Design and Analysis of Algorithms</td><td>Regulation:</td><td>2018</td></tr>
<tr><td>Academic Year</td><td>2021-2022</td><td>Batch</td><td>2020-2024</td><td>Date</td><td>29.03.2022</td><td>FN</td></tr>
<tr><td>Time: 90 Minutes</td><td colspan="4">Answer All Questions</td><td colspan="2">Maximum: 50 Marks</td></tr>
</table>

Part – A (6×2 = 12 Marks)

<table>
<tr><td>K2</td><td>1. f(n)=n^a, g(n)=f(2n). Compute the order of growth of g(n) compared to f(n)?
2^a(n^a) /n^a</td><td>CO3</td><td>1.1.1, 2.1.1</td></tr>
<tr><td>K2</td><td>2. Mention the lower and upper bounds of an algorithm for finding whether a number n is prime or not.</td><td>CO1</td><td>2.1.1, 2.4.3</td></tr>
<tr><td>K1</td><td>3. List any two properties of asymptotic notations
If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, then
$$t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\}).$$
$$\left.\begin{array}{|c|}\hline t_1(n) \in O(g_1(n)) \\ \hline t_2(n) \in O(g_2(n)) \\ \hline \end{array}\right\} \quad t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\}).$$</td><td>CO2</td><td>1.3.1</td></tr>
<tr><td>K2</td><td>4. Mention the lower and upper bounds for the total number of swaps of bubble sort algorithm?
<table>
<tr><th>Name</th><th>Average</th><th>Worst</th><th>Memory</th><th>Stable</th><th>Method</th></tr>
<tr><td>Bubble sort</td><td>$O(n^2)$</td><td>$O(n^2)$</td><td>$O(1)$</td><td>Yes</td><td>Exchanging</td></tr>
<tr><td>Selection sort</td><td>$O(n^2)$</td><td>$O(n^2)$</td><td>$O(1)$</td><td>No</td><td>Selection</td></tr>
<tr><td>Insertion sort</td><td>$O(n^2)$</td><td>$O(n^2)$</td><td>$O(1)$</td><td>Yes</td><td>Insertion</td></tr>
<tr><td>Merge sort</td><td>$O(n \log n)$</td><td>$O(n \log n)$</td><td>$O(n)$</td><td>Yes</td><td>Merging</td></tr>
<tr><td>Quicksort</td><td>$O(n \log n)$</td><td>$O(n^2)$</td><td>$O(1)$</td><td>No</td><td>Partitioning</td></tr>
<tr><td>Heapsort</td><td>$O(n \log n)$</td><td>$O(n \log n)$</td><td>$O(1)$</td><td>No</td><td>Selection</td></tr>
</table></td><td>CO2</td><td>2.1.1, 2.4.3</td></tr>
<tr><td>K2</td><td>5. Calculate the number of comparison i >= n performed in the following program?
int i = 200, n = 80;</td><td>CO1</td><td>2.1.3</td></tr>
</table>

	main() *{* *while (i >= n)* *{* *i = i-2;* *n = n+1;* *}* *}* *42*		
K2	6. Organize the following functions in increasing order of asymptotic growth: nlogn, n^2 , logn, n, n!, n^3 . logn, n, nlogn,n^2, n^3,n!	CO2	1.3.1, 1.4.1

Part – B (3×6 = 18 Marks)

K2	7. Explain the motivations for the analysis of algorithms Algorithm analysis is an important part of computational complexity theory, which provides theoretical estimation for the required resources of an algorithm to solve a specific computational problem. Analysis of algorithms is the determination of the amount of time and space resources required to execute it. Performance: How much time/memory/disk/etc. is used when a program is run. This depends on the machine, compiler, etc. as well as the code we write. Complexity: How do the resource requirements of a program or algorithm scale, i.e. what happens as the size of the problem being solved by the code gets larger.	CO3	1.4.1
K3	8. Consider the following functions and identify the one whose growth is faster, $f(n) = \sqrt{n}.n!$ and $g(n) = \sqrt{n}.2^n$	CO3	2.4.1, 13.3.1
K2	9. Summarize Master's theorem for dividing recursive functions **Master Theorem** If $f(n) \in \Theta(n^d)$ where $d \geq 0$ in recurrence (5.1), then $T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d, \\ \Theta(n^d \log n) & \text{if } a = b^d, \\ \Theta(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$ Analogous results hold for the O and Ω notations, too.	CO2	1.1.1, 2.1.3

Part – C (2×10 = 20 Marks)

K3	10. Explain Asymptotic notations in detail. Compute *O( )*, *Θ( )* and *Ω( )* for Selection sort and Linear Search algorithms
	ALGORITHM *SelectionSort*$(A[0..n-1])$
	//Sorts a given array by selection sort
	//Input: An array $A[0..n-1]$ of orderable elements
	//Output: Array $A[0..n-1]$ sorted in nondecreasing order
	for $i \leftarrow 0$ **to** $n-2$ **do**
	$min \leftarrow i$
	for $j \leftarrow i+1$ **to** $n-1$ **do**
	if $A[j] < A[min]$ $min \leftarrow j$
	swap $A[i]$ and $A[min]$
	$C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [(n-1) - (i+1) + 1] = \sum_{i=0}^{n-2} (n-1-i).$
		CO1	2.1.1, 2.4.3

	ALGORITHM *SequentialSearch2*$(A[0..n], K)$
	//Implements sequential search with a search key as a sentinel
	//Input: An array A of n elements and a search key K
	//Output: The index of the first element in $A[0..n-1]$ whose value is
	// equal to K or -1 if no such element is found
	$A[n] \leftarrow K$
	$i \leftarrow 0$
	while $A[i] \neq K$ **do**
	$i \leftarrow i + 1$
	if $i < n$ **return** i
	else return -1		

(OR)			
K3	11. Derive time complexity for Binary search in which insertion of elements happens one by one through a Binary Search Tree data structure. Analyze Best case, Worst case, and average case for the binary search algorithm on BST	CO1	2.1.1, 2.4.3
K4	12. Analyze the following code snippet, setup the recurrence relation and derive its Time complexity. *def fun(n):* *if (n==1)* *return 1* *else* *for (i=1;i<n;i=i*2);* *{* *//do something in O(1)* *}* *fun(n/2)*	CO2	2.1.1, 2.1.3, 2.4.1
(OR)			
K4	13. Analyze the following code snippet, setup the recurrence relation and derive its Time complexity. *def fun(n):* *if (n==0)* *return 1* *else* *for(i=1;i<n;i=i+1;* *{* *//do something in O(1)* *}* *fun(n-3)* *fun(n-3)* *fun(n-3)*	CO2	2.1.1, 2.1.3, 2.4.1

Prepared By	Reviewed By	Approved By
Course Coordinator	PAC Team	HOD

Guidelines

1. The question paper should be set in accordance with Bloom's Taxonomy (APPENDIX – A: next page). The questions in a desired knowledge level must contain the respective action verbs.
2. The Knowledge level (Eg. <K2>), course outcome (Eg. <CO2>), and the program indicators (Eg. <1.2.1>) should be mentioned against each question and subdivisions in the respective columns.
3. Both the questions in "either or" type must be set in the same knowledge level and must be from the same CO.
4. In the case of "either or" type questions, the keyword (OR) must be in a separate row.
5. In the case of sub-divisions in a question, it is preferable to have the same knowledge level.
6. The marks assigned to each question in the case of subdivisions should be mentioned clearly at the end of the question within brackets and with a keyword Marks. (Eg. (5 Marks)).
7. Add the keyword "Options" before the choices of an objective type question in Part A.
8. Once the question paper is set, its adherence to the guidelines in terms of knowledge levels and marks distribution has to be approved by the QP Scrutiny Team.

APPENDIX – A
Bloom's Taxonomy Action Verbs

K Level	Bloom's Definition	Action Verbs
K1 Remember	Exhibit memory of Previously learned Material by recalling facts, terms, basic concepts, and answers.	Choose, Define, Find, How, Label, List, Match, Name, Omit, Recall, Relate, Show, Spell, Tell, What, When, Where, Which, Who, Why.
K2 Understand	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions and stating main ideas.	Classify, Compare, Contrast, Demonstrate, Explain, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate
K3 Apply	Solve problems to new situations by applying acquired knowledge, facts, techniques, and rules in a different way.	Apply, Build, Construct, Develop, Experiment with, Identify, Interview, Make use of, Model, Organize, Plan, Select, Solve, Utilize
K4 Analyse	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support generalizations	Analyze, Assume, Categorize, Conclusion, Discover, Dissect, Distinguish, Divide, Examine, Function, Inference, Inspect, Motive, Relationships, Simplify, Survey, Take part in, Test for, Theme
K5 Evaluate	Present and defend opinions by making judgments about information, validity of ideas, or quality of work based on a set of criteria.	Agree, Appraise, Assess, Award, Choose, Compare, Conclude, Criteria, Criticize, Decide, Deduct, Defend, Determine, Disprove, Estimate, Evaluate, Explain, Importance, Influence, Interpret, Judge, Justify, Mark, Measure, Opinion, Perceive, Prioritize, Prove, Rate, Recommend, Rule on, Select, Support, Value
K6 Create	Compile information together in a different way by combining	Adapt, Build, Change, Choose, Combine, Compile, Compose, Construct, Create,

	elements in a new pattern or proposing alternative solutions.	Delete, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Happen, Imagine, Improve, Invent, Make up, Maximize, Minimize, Modify, Original, Originate, Plan, Predict, Propose, Solution, Solve, Suppose, Test, Theory