National University of Computer & Emerging Sciences, Karachi

Fall-2021 Department of Computer Science

Final Exam: Part (A)

29 December 2021, 09:00 AM – 12:00 AM

Course Code: CS2009	**Course Name:** Design and Analysis of Algorithm
Instructor Name / Names: Dr. Muhammad Atif Tahir, Dr. Fahad Sherwani, Dr. Farrukh Saleem, Waheed Ahmed, Waqas Sheikh, Sohail Afzal	
Student Roll No:	**Section:**

Instructions:

- Must be submitted within 30 minutes, you are allowed to submit the paper before 30 minutes, and start Part(B)
- No extra sheets are allowed for this section. You must solve the questions in provided space on the question paper.

Time: 30 minutes **Max Marks: 10**

Question # 1 **[0.5*6 = 3 marks]**

Answer the following questions. You must explain in only 3-4 lines.

(a) List the following functions according to their order of growth from the lowest to the highest.

$(n-2)!$, $5\log(n+100)^{10}$, 2^{2n}, $0.001\,n^4+3n^3+1$, $\ln^2 n$, $\sqrt[3]{n}$, 3^n.

(b) Explain why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

(c) Define recurrence relations and enlist methods to solve them.

(d) In which conditions dynamic programing does not work. Give suitable example.

(e) Describe Big theta in mathematical notation.

(f) Suppose there is a maximization problem, where the approximate solution has the cost of 25 And optimal solution has the cost of 30. Find the approximation ratio.

Question # 2 **[0.5*8 = 4 marks]**

Write the complexity and the corresponding design strategy (Divide and Conquer/ Dynamic Programming / Greedy) of the given algorithms

ALGORITHMS	**Worst Case**	**Design Strategy**
Quick Sort		
Radix Sort		
Max-Heapify operation		
Add Vertex in Adjacency Metric		
Rod-Cutting (Dynamic programming)		
Dijkstra's (using Array)		
Prims		
Maximum Sub-array Sum		

Question # 3 **[1*3 = 3 marks]**

For each of the following questions indicate whether it is true or false and justify using some examples by assuming a function.

(a) For all positive $f(n)$; $\omega(f(n)) + O(f(n)) = \Theta(f(n))$

(b) For all positive $f(n)$; $f(n) + o(f(n)) = \Theta(f(n))$

(c) For all positive $f(n), g(n), and\ h(n)$: if $f(n) = O(g(n))\ and\ f(n) = \Omega(h(n))\ then\quad g(n) + h(n) = \Omega(f(n))$