BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI, DUBAI CAMPUS
DUBAI INTERNATIONAL ACADEMIC CITY, DUBAI
SECOND SEMESTER 2024 – 2025
DEPARTMENT OF COMPUTER SCIENCE

A

COURSE NAME : Design & Analysis of Algorithms
COMPONENT : Quiz – I (Open Book)
DATE/DAY : 18th March 2025, Tuesday

COURSE CODE : CS F364
WEIGHTAGE : 15%
DURATION : 30 mins

Instructions:

1. Read the questions carefully before attempting.
2. Write down any assumptions you make.
3. Show all steps to score full marks.

Reg. No : ______________________
Name : ______________________
Section : ______________________

Answer all the Questions

PART -A (5 X 2 = 10 Marks)

No Step/Partial Marks. Show whether complexity relations with respect to O, Ω, Θ are true/ false for each pair

1. Compare the following pairs of functions in terms of order of magnitude. For each pair, clearly state whether it is true/false with respect to these complexity relations(all three conditions):

$$f(n) = O(g(n))$$

$$f(n) = \Omega(g(n))$$

$$f(n) = \Theta(g(n))$$

a) $f(n) = 100n + \log n, g(n) = n + (\log n)^2$

Dominant terms: Both have leading terms linear in n. Thus, both are dominated by n. Hence, these two functions grow at the same rate.

- $f(n) = O(g(n))$: ✅ TRUE
- $f(n) = \Omega(g(n))$: ✅ TRUE
- $f(n) = \Theta(g(n))$: ✅ TRUE

b) $f(n) = \frac{n^2}{\log n}, g(n) = n(\log n)^2$

$f(n) = \frac{n^2}{\log n}$ grows faster than $g(n) = n(\log n)^2$ because n^2 dominates $n(\log n)^2$ for large n

- $f(n) = O(g(n))$: ❌ FALSE
- $f(n) = \Omega(g(n))$: ✅ TRUE
- $f(n) = \Theta(g(n))$: ❌ FALSE

c) $f(n) = (\log n)^{\log n}, g(n) = \frac{n}{\log n}$

$(\log n)^{\log n}$ grows faster than $\frac{n}{\log n}$ because $(\log n)^{\log n}$ is exponential in $\log n$, while $\frac{n}{\log n}$ is polynomial.

- $f(n) = O(g(n))$: ❌ FALSE
- $f(n) = \Omega(g(n))$: ✅ TRUE
- $f(n) = \Theta(g(n))$: ❌ FALSE

d) $f(n) = \sqrt{n}, g(n) = (\log n)^5$

Clearly polynomial $\sqrt{n}$ grows faster than poly-logarithmic $(\log n)^5$

- $f(n) = O(g(n))$: ❌ FALSE
- $f(n) = \Omega(g(n))$: ✅ TRUE
- $f(n) = \Theta(g(n))$: ❌ FALSE

e) $f(n) = n2^n, g(n) = 3^n$

3^n grows faster than $n2^n$ because the base of the exponential term $(3 > 2)$ dominates

- $f(n) = O(g(n))$: ✅ TRUE (since $f(n)$ grows slower)
- $f(n) = \Omega(g(n))$: ❌ FALSE
- $f(n) = \Theta(g(n))$: ❌ FALSE

PART -B (10 Marks)

1. In the Master's Theorem, which case applies if $f(n) = O(n^c)$ where $c < \log_b a$? [1]
   a) Case 1: Divide dominates
   b) Case 2: Work done at each level is the same
   c) Case 3: Combine dominates
   d) No case applies

Answer: a) Case 1: Divide dominates

2. Which of the following is NOT a characteristic of the recursion tree method? [1]
   a) It helps in visualizing recursive calls
   b) It provides an exact solution to the recurrence
   c) It shows the total work done at each level
   d) It can be used to derive an asymptotic bound

Answer: b) It provides an exact solution to the recurrence

3 Which of the following statements are true: [3]

   a) In a recursion tree, each level represents a recursive call with reduced input size.
   b) The stopping condition in a recursion tree occurs when a certain number of levels are reached.
   c) In the recurrence $T(n) = aT\left(\frac{n}{b}\right) + f(n)$, the term a represent the number of subproblems.
   d) If $f(n)$ grows faster than $n^{\log_b a}$ in the recurrence $T(n) = aT\left(\frac{n}{b}\right) + f(n)$, case 2 of master's theorem applies.
   e) The time complexity of the recurrence $T(n) = 0.9T\left(\frac{n}{2}\right) + O(\frac{1}{n})$ using master's theorem is O(1/n).
   f) $T(n) = T\left(\frac{n}{2}\right) + k$, represents the recurrence equation for binary search.

Answer: a) True b) False c) True d) False e) False d) True

4 What is the time complexity of the recurrence $T(n) = 3^n T\left(\frac{n}{3}\right) + n^n$ using master's theorem? [1]

Answer: Can't be solved using master's theorem

5 What is the time complexity of the recurrence $T(n) = 9T(\frac{n}{2}) + O(n^2)$ using recursion tree method, where T (1) =1? Also, what will be the work done at level k+1 (if we are starting from level 1)? [4]

Answer: $\boldsymbol{O(n^{3.17})\ or\ O\left(n^{2+\log_2 \frac{9}{4}}\right) or\left(n^{\log_2 9}\right)}$ **and** $\boldsymbol{9^k(n/2^k)^2}$

PART – C (10 X 1 = 10 Marks)

1 Which of the following is not a characteristic of divide and conquer approach?

a) Divide
b) Combine
c) Optimal substructure
d) Conquer
e) None of the above

Ans: Option e

2 The minimum number of comparisons required to find the minimum and the maximum of n numbers using divide and conquer approach.

a) $\frac{n}{2} - 2$
b) $\frac{3n}{2}$
c) $\frac{3n}{2} - 2$
d) $n - 1$

Ans: Option c

3 What is the worst-case number of arithmetic operations performed by binary search on a sorted array of size n?

a) $log_{\,2}(n)$
b) n
c) $n - 1$
d) $log_{\,10}(n)$

Ans: Option a

4. Let you have written a program to implement QuickSort to sort numbers in ascending order. Assume that you are using first element as pivot. Now, you are testing the program for different inputs based on the number of comparisons made by the QuickSort. Let t1 and t2 be the number of comparisons made by the QuickSort for the inputs {11, 22, 33, 44, 55, 66} and {44, 11, 55, 33, 22, 66} respectively. Which one of the following holds?

a) $t_1 > t_2$
b) $t_1 < t_2$
c) $t_1 == t_2$
d) Cannot compare t_1 and t_2

Ans: Option a

5. Suppose you have a procedure to find the kth minimum element from a given array in constant time. You use this kth element as pivot in Quick Sort then what will be recurrence relation for the algorithm.

a) $T(n) = T(n-1) + n$
b) $T(n) = T(n-k) + n$

c) $T(n) = T(n-k) + T(k) + n$
d) $T(n) = T\left(\frac{n}{k}\right) + n$

Ans: Option c

6. Given two sorted list of size m and n respectively. The number of comparisons needed in worst case by the merge sort algorithm is
   a) $m * n$
   b) maximum of m and n
   c) minimum of m and n
   d) $m + n - 1$

Ans: Option d

7 What will be the maximum possible array size at the i^{th} level of the merge sort tree for an array of size N (assume level indexing starts from 0, i.e., the original array at the root is said to be at the 0^{th} level)? Fractional results will be rounded up to the nearest integer.
   a) $2^i * n$
   b) $i * n$
   c) $\frac{n}{i}$
   d) $\frac{n}{2^i}$

Ans: Option d

8 Imagine a modified version of the binary search called "Weighted Binary Search." Unlike the standard binary search that always checks the middle element of an array, this method chooses the ⌊n*0.70⌋ or ⌊n*0.30⌋ as pivot then what can be the best- and worst-case time complexity.
   a) $log_{0.7}(\frac{1}{n})\ and\, log_{0.3}(\frac{1}{n})$
   b) $log_{0.7}(n)\ and\, log_{0.3}(n)$
   c) $log_2(\frac{1}{n})\ and\, log_2(\frac{1}{n})$
   d) $log_2(n)\ and\, log_2(n)$

Ans: Option a

9. Which of the following algorithm is not a in-place sorting algorithm. Hints: A sorting algorithm is said to be in-place if it doesn't use separate array during sorting.
   a) Quick sort
   b) Merge sort
   c) Insertion sort
   d) Selection sort

Ans: Option b

10 Which of the following algorithm is not a stable sorting algorithm. Hints: An algorithm is said to be stable if it does not swap to identical element during sorting.
   a) Quick sort
   b) Merge sort
   c) Insertion sort
   d) Selection sort

Ans: Option c

********** ALL THE BEST *********