Indian Institute of Science Education and Research Thiruvananthapuram
Data Structures (DSC 314/MAT 5514)
Quiz 1 (February 3, 2026)

Time: 30 min Max. marks: 20

Answer all questions. Each question carries 2 marks.

1 For each of the following pieces of code, find the number of times **op()** is called as a function of the input size n. Express your answer in terms of the **Big-Theta** notation. Also, justify your answer.

(a) for $(i = 10; i < n + 5; i+ = 2)$
op();

Answer: $\theta(n)$
Justification: The loop starts at $i = 10$ and increases by 2 each time until it reaches $n + 5$. The number of iterations is approximately

$$\frac{(n+5) - 10}{2} = \frac{n-5}{2}.$$

Ignoring constants, this grows linearly with n. Therefore, the number of times **op()** is called is $\Theta(n)$.

(b) for $(i = 1; i < n; i* = 2)$
op();

Answer: $\Theta(\log n)$
Justification: The loop starts at $i = 1$ and doubles each time: $1, 2, 4, 8, \ldots$. After k iterations, $i = 2^k$. The loop stops when $2^k \geq n$. Solving,

$$2^k \geq n \Rightarrow k \geq \log_2 n.$$

Hence, the number of times **op()** is called is $\Theta(\log n)$.

2 Express the function $f(n) = (n^3)/1000 - 100n^2 - 100n + 3$ in terms of θ notation.

Answer: $\theta(n^3)$

3 Which of the following functions provides the maximum asymptotic complexity?

a) $f(n) = n^{(3/2)}$ b) $f(n) = n^{(5/4)}$ c) $f(n) = nlogn$ d) $f(n) = 2^n$
Answer: $f(n) = 2^n$

4 Which of the following is a Divide and Conquer algorithm?

a) Bubble Sort b) Selection Sort c) Insertion Sort d) Quick Sort

Answer: Quick Sort

5 Which of the following sorting algorithms provides the best time complexity in the worst-case scenario?

a) Bubble Sort b) Merge Sort c) Insertion Sort d) Quick Sort

Answer: Merge Sort

6 Identify the sorting technique that compares adjacent elements in a list and switches whenever necessary.

a) Bubble Sort b) Merge Sort c) Insertion Sort d) Quick Sort

Answer: Bubble Sort

7 Among the following options which is the best sorting algorithm when the list is already sorted?

a) Bubble Sort b) Merge Sort c) Insertion Sort d) Quick Sort

Answer: Insertion Sort (bubble sort takes $O(n^2)$ unless it is optimized with a swap-check flag.)

8 What is the best case time complexity of the binary search algorithm? Justify your answer.

a) $\mathcal{O}(1)$ b) $\mathcal{O}(n)$ c) $\mathcal{O}(logn)$ d) $\mathcal{O}(n^{(3/2)})$

Answer: The best case time complexity of a binary search algorithm is $\mathcal{O}(1)$, which means constant time; this occurs when the target element is found at the middle index of the array, requiring only one comparison to identify it.

9 Show that the running time of Quick Sort is $\mathcal{O}(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

Answer: If the array is already sorted in decreasing order, then, the pivot element is less than all the other elements. The partitioning step takes $\theta(n)$ time, and then leaves with a subproblem of size $n-1$ and a subproblem of size 0. This gives us the recurrence $T(n) = T(n-1)+T(0)+\theta(n)$. Note that: $T(0) = \theta(1)$. So, in each level we will have n decreased by 1 and the partitioning cost is still $\theta(n)$, that leaves us with $\mathrm{T}(n-1) = T(n-2)+T(0)+\theta(n)$ again. Thus $T(n) = \mathcal{O}(n^2)$.

10 Solve the recurrence:

$T(n) = T(n-1)+1, \qquad T(1) = 1$

Answer:

$$T(n) = T(n-1)+1$$

$$
\begin{aligned}
T(n) &= T(n-1) + 1 \\
&= (T(n-2) + 1) + 1 \\
&= T(n-2) + 2 \\
&= T(n-3) + 3 \\
&\vdots \\
&= T(1) + (n-1)
\end{aligned}
$$

$$T(n) = T(1) + (n-1)$$

$$T(n) = \Theta(n)$$