

✓ LEVEL 1 – INTERVIEW QUESTIONS WITH ANSWERS

1 Fibonacci Series

Q1. What is the time complexity of your solution?

☞ **Answer:**

$O(n)$ because each Fibonacci number is computed once in a loop.

Q2. Why not use recursion?

☞ **Answer:**

Recursive Fibonacci has exponential time complexity $O(2^n)$ due to repeated calls, so it is not optimal.

Q3. Space complexity of your solution?

☞ **Answer:**

$O(1)$ because I only use a few variables.

Q4. What happens for $n = 0$ or $n = 1$?

☞ **Answer:**

I handle them as base cases:

$n = 0 \rightarrow$ no output, $n = 1 \rightarrow 0$

2 Factorial

Q1. Why is $0! = 1$?

☞ **Answer:**

Because factorial represents the number of ways to arrange zero objects, which is 1 by definition.

Q2. What about negative numbers?

☞ **Answer:**

Factorial is not defined for negative numbers, so I return an error or message.

Q3. Iterative vs Recursive factorial?

☞ **Answer:**

Iterative is preferred because recursion adds extra stack space.

Q4. Time & space complexity?

☞ **Answer:**

Time → $O(n)$

Space → $O(1)$

3 Sum of Elements in Array

Q1. Can you find the sum without storing the array?

☞ **Answer:**

Yes, by adding elements as they are read, but the question specifically asks for an array.

Q2. Time & space complexity?

☞ **Answer:**

Time → $O(n)$

Space → $O(1)$ (excluding input array)

Q3. Does it work for negative numbers?

☞ **Answer:**

Yes, addition works for all integers.

4 Linear Search

Q1. Time complexity?

☞ **Answer:**

Best case → $O(1)$

Worst case → $O(n)$

Q2. When should linear search be used?

☞ **Answer:**

When the array is unsorted or very small.

Q3. What if the element occurs multiple times?

☞ **Answer:**

I can either return the first index or continue searching for all occurrences.

5 Frequency of Elements (map / unordered_map)

Q1. Difference between map and unordered_map?

☞ Answer:

map stores elements in sorted order and takes $O(\log n)$ time.
unordered_map uses hashing and takes $O(1)$ average time.

Q2. Worst case of unordered_map?

☞ Answer:

Worst case is $O(n)$ due to hash collisions.

Q3. Java & Python equivalent?

☞ Answer:

Java → HashMap

Python → dict

6 Palindrome String

Q1. Optimal approach?

☞ Answer:

Two-pointer technique comparing characters from both ends.

Q2. Time & space complexity?

☞ Answer:

Time → $O(n)$

Space → $O(1)$

Q3. How to ignore spaces and case?

☞ Answer:

Convert to lowercase and skip non-alphanumeric characters.

7 Palindrome Number

Q1. Are negative numbers palindrome?

☞ Answer:

No, because of the negative sign.

Q2. Why not convert to string?

☞ **Answer:**

Math-based solution is more optimal and avoids extra space.

Q3. Time & space complexity?

☞ **Answer:**

Time → $O(d)$

Space → $O(1)$

8 Reverse Array

Q1. Difference between printing reverse and reversing array?

☞ **Answer:**

Printing reverse does not change the array, reversing modifies it in-place.

Q2. Optimal solution?

☞ **Answer:**

Two-pointer approach.

Q3. Time & space complexity?

☞ **Answer:**

Time → $O(n)$

Space → $O(1)$

9 Prime Number

Q1. Why check till \sqrt{n} ?

☞ **Answer:**

If $n = a \times b$, at least one factor must be $\leq \sqrt{n}$.

Q2. Time & space complexity?

☞ **Answer:**

Time → $O(\sqrt{n})$

Space → $O(1)$

Q3. How to check primes for many numbers?

☞ **Answer:**

Use Sieve of Eratosthenes.

[10] Maximum Element in Array

Q1. Why initialize max with arr[0]?

 **Answer:**

Because array may contain negative numbers.

Q2. Time & space complexity?

 **Answer:**

Time $\rightarrow \mathcal{O}(n)$

Space $\rightarrow \mathcal{O}(1)$

Q3. Can this be done in one traversal?

 **Answer:**

Yes, and that is what I implemented.
