■ Java Interview Revision Guide

A quick, visual-friendly and easy-to-understand reference before interviews.

■ 1. Reverse a String

■ **Logic:** Logic: Convert the string to a character array and swap characters from both ends moving inward. Alternatively, use the built-in StringBuilder.reverse().

■ **Visual:**

Diagram: [s] h e l l o → swap h↔o → o e l l h → ‘olleh’

■ **Easy Explanation:** Explanation: You use two pointers (start and end). Keep swapping until they meet in the middle.

■ 2. Palindrome Check

■ **Logic:** Logic: Compare characters from start and end of string; if all pairs match → palindrome.

■ **Visual:**

Example: madam → matches m↔m, a↔a → palindrome.

■ **Easy Explanation:** Explanation: Two-pointer technique ensures O(n) time and no extra space.

■ 3. Prime Number

■ **Logic:** Logic: Number greater than 1 and divisible only by 1 and itself. Check divisibility up to √n.

■ **Visual:**

Diagram: For 9 → check 2,3 → 9 % 3 == 0 → not prime.

■ **Easy Explanation:** Explanation: Checking till √n avoids redundant checks and improves efficiency.

■ 4. Factorial

■ **Logic:** Logic: Multiply numbers from 1 to n → n!

■ **Visual:**

Example: 5! = 1×2×3×4×5 = 120

■ **Easy Explanation:** Explanation: Iterative approach avoids recursion overhead. Useful in permutations.

■ 5. Fibonacci Series

■ **Logic:** Logic: Each term is the sum of the previous two terms.

■ **Visual:**

Example: 0,1,1,2,3,5,8,…

■ **Easy Explanation:** Explanation: Iterative version avoids recursion overhead and runs in O(n) time.

■ Time Complexity Summary

Algorithm	Time	Space	Stable
Linear Search	O(n)	O(1)	–
Binary Search	O(log n)	O(1)	–
Bubble Sort	$O(n^2)$	O(1)	Yes
Selection Sort	$O(n^2)$	O(1)	No
Insertion Sort	$O(n^2)$	O(1)	Yes
Merge Sort	O(n log n)	O(n)	Yes
Quick Sort	O(n log n)	O(log n)	No