Top 10 Array Interview Questions in Java

1) Largest & Smallest element

Idea: scan once, track min and max. Edge cases: all negative; all equal; single element. Time: O(n) • Space: O(1).

2) Reverse an array (in-place)

Idea: two pointers (l, r) swapping until they cross. Edge cases: empty array; odd length. Time: O(n) • Space: O(1).

3) Check if array is sorted (ascending)

Idea: ensure every a[i] <= a[i+1]. Edge cases: duplicates allowed; length 0/1 is “sorted”. Time: O(n) • Space: O(1).

4) Second largest (distinct)

Idea: track top two distinct values in one pass. Edge cases: not enough distinct values (e.g., {7,7}) → return “not found”. Time: O(n) • Space: O(1).

5) Find duplicates (unsorted)

Idea: use a HashSet to detect repeats; store them in another set. Edge cases: no duplicates; many duplicates of same value. Time: O(n) • Space: O(n).

6) Remove duplicates from a sorted array (in-place)

Idea: two-pointer “write index” technique; return new logical length. Edge cases: all unique; all identical; empty. Time: O(n) • Space: O(1).

7) Missing number from 1..N (one missing)

Idea: XOR from 1..N with XOR of array (avoids overflow). Edge cases: missing first or last. Time: O(n) • Space: O(1).

8) Rotate array right by k

Idea: 3-reversals trick: reverse all, reverse first k, reverse remaining. Edge cases: $k \geq n \rightarrow$ use k %= n; k = 0. Time: O(n) • Space: O(1).

9) Pairs with a given sum

Idea: use a HashSet for seen values and a Set to avoid duplicate pairs. Edge cases: repeated numbers; negative values. Time: O(n) • Space: O(n).

10) Majority element (> n/2 times)

Idea: Boyer–Moore voting to get candidate, then verify count. Edge cases: no majority present. Time: O(n) • Space: O(1).