

# 55 Data Structures Question Bank

1. Reverse an array in-place.

Example: Input: [1, 2, 3, 4, 5] → Output: [5, 4, 3, 2, 1]

2. Perform linear search and return the index of a key if found.

Example: Array: [10, 20, 30, 40], Key: 30 → Output: Index 2

3. Perform binary search on a sorted array and return the index of the element.

Example: Array: [1, 3, 5, 7, 9], Key: 5 → Output: Index 2

4. Rotate an array to the right by k steps.

Example: Array: [1, 2, 3, 4, 5], k = 2 → Output: [4, 5, 1, 2, 3]

5. Merge two sorted arrays without using extra space.

Example: A: [1, 3, 5], B: [2, 4, 6] → Output: [1, 2, 3, 4, 5, 6]

6. Implement selection sort.

Example: Array: [64, 25, 12, 22, 11] → Output: [11, 12, 22, 25, 64]

7. Implement bubble sort and optimize it using a flag.

Example: Input: [5, 1, 4, 2, 8] → Output: [1, 2, 4, 5, 8]

8. Implement insertion sort.

Example: Input: [9, 5, 1, 4, 3] → Output: [1, 3, 4, 5, 9]

9. Sort an array using merge sort (recursive).

Example: Input: [6, 3, 9, 5, 2] → Output: [2, 3, 5, 6, 9]

10. Sort an array using quick sort (Lomuto partition).

Example: Input: [10, 80, 30, 90, 40] → Output: [10, 30, 40, 80, 90]

11. Find the second largest element in an array.

Example: Input: [12, 35, 1, 10, 34, 1] → Output: 34

12. Find the missing number in array from 1 to  $n+1$ .

Example: Input: [1, 2, 4, 6, 3, 7, 8] → Output: 5

13. Find the maximum subarray sum (Kadane's algorithm).

Example: Input: [-2,1,-3,4,-1,2,1,-5,4] → Output: 6

14. Count occurrences of a key in sorted array.

Example: Input: [1, 2, 2, 2, 3], Key: 2 → Output: 3

15. Find pivot index in a sorted rotated array.

Example: Input: [4, 5, 6, 7, 0, 1, 2] → Output: Index 4

16. Find index of an element in a sorted rotated array.

Example: Array: [4,5,6,7,0,1,2], Key: 6 → Output: Index 2

17. Implement singly linked list with insert at head and tail.

18. Delete a node at a given position in a singly linked list.

Example: List:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , Position: 2  $\rightarrow$  Output:  $1 \rightarrow 2 \rightarrow 4$

19. Reverse a singly linked list iteratively.

Example: Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow$  Output:  $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$

20. Detect loop in a linked list using Floyd's Algorithm.

21. Find the middle element of a singly linked list.

Example: Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow$  Output: 3

22. Check if a singly linked list is a palindrome.

Example: Input:  $1 \rightarrow 2 \rightarrow 2 \rightarrow 1 \rightarrow$  Output: Yes

23. Merge two sorted linked lists.

Example: L1:  $1 \rightarrow 3 \rightarrow 5$ , L2:  $2 \rightarrow 4 \rightarrow 6 \rightarrow$  Output:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$

24. Implement a stack using array.

25. Implement a stack using linked list.

- 26. Implement push, pop, peek operations on stack.
- 27. Check for balanced parentheses using stack.

Example: Input: "{[()]}" → Output: Balanced

- 28. Sort a stack using recursion.

Example: Input: [3, 1, 4, 2] → Output: [1, 2, 3, 4]

- 29. Implement a queue using array.
- 30. Implement a circular queue using array.
- 31. Implement a queue using linked list.
- 32. Implement a queue using two stacks.
- 33. Implement a stack using two queues.
- 34. Evaluate postfix expression using stack.

Example: Input: "231\*+9-" → Output: -4

- 35. Convert infix to postfix using stack.

Example: Input: (A+B)C → Output: AB+C

36. Implement a priority queue using array.

37. Find next greater element for each element in the array.

Example: Input: [4, 5, 2, 25] → Output: [5, 25, 25, -1]

38. Implement doubly linked list with insert and delete.

39. Reverse a doubly linked list.

Example: Input:  $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5 \rightarrow$  Output:  $5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1$

40. Implement a dequeue (double-ended queue).

41. Move all zeros to the end without changing order.

Input: [0, 1, 0, 3, 12] → Output: [1, 3, 12, 0, 0]

42. Find if any pair exists with a given sum.

Input: [10, 15, 3, 7], Sum = 17 → Output: Yes (10 + 7)

43. Remove duplicates from a sorted array.

Input: [1, 1, 2, 3, 3] → Output: [1, 2, 3]

44. Find the equilibrium index in an array.

Input: [-7, 1, 5, 2, -4, 3, 0] → Output: Index 3

45. Find the floor of a number in a sorted array.

Array: [1, 2, 8, 10, 10, 12, 19], x = 5 → Output: 2

46. Find first and last occurrence of a key in a sorted array.

Array: [2, 4, 4, 4, 6, 7], key = 4 → Output: 1 and 3

47. Count the number of inversions in an array.

Input: [2, 4, 1, 3, 5] → Output: 3 inversions

48. Check if the array is already sorted.

Input: [1, 2, 3, 4, 5] → Output: Yes

49. Remove duplicates from a sorted linked list.

Input: 1 → 1 → 2 → 3 → 3 → Output: 1 → 2 → 3

50. Swap nodes in pairs in a linked list.

Input: 1 → 2 → 3 → 4 → Output: 2 → 1 → 4 → 3

51. Find intersection point of two linked lists.

(Given they merge at a node)

52. Design a special stack with getMin() in O(1).

53. Find the span of stock prices using stack.

Input: [100, 80, 60, 70, 60, 75, 85] → Output: [1,1,1,2,1,4,6]

54. Find the first non-repeating character in a stream.

Input: a a b c → Output: a – b b

55. Generate binary numbers from 1 to N using queue.

Input: N = 5 → Output: 1, 10, 11, 100, 101

---