DATA STRUCTURES

ARRAYS – MEDIUM LEVEL

1. Which of the following operations has the worst-case time complexity of O(n) in an array?
   A) Accessing an element by index
   B) Insertion at the end
   C) Deletion from the beginning
   D) Traversal
   Answer: C

2. If an array has base address B, element size w, and is 1-dimensional with index i, the address of A[i] is:
   A) B + i
   B) B + i * w
   C) B + w
   D) B + (i-1)
   Answer: B

3. Which of these is not possible with arrays in most languages?
   A) Random access
   B) Changing array size dynamically
   C) Sequential traversal
   D) Storing primitive data types
   Answer: B

4. What is the worst-case time complexity of searching an unsorted array?
   A) O(1)
   B) O(log n)
   C) O(n)
   D) O(n log n)
   Answer: C

5. The space complexity of an array is:
   A) O(1)
   B) O(log n)
   C) O(n)
   D) $O(n^2)$
   Answer: C

6. Which data structure is more suitable for frequent insertions and deletions at random positions?
   A) Array
   B) Linked List
   C) Queue
   D) Stack
   Answer: B

7. If an array is sorted, the best searching algorithm is:
   A) Linear search
   B) Binary search
   C) Jump search
   D) Hashing
   Answer: B

8. What is the time complexity of inserting an element into a sorted array (worst case)?
   A) O(1)
   B) O(log n)
   C) O(n)
   D) O(n log n)
   Answer: C

9. Which of these is true about multi-dimensional arrays?
   A) They can only be stored in contiguous memory
   B) They require nested loops to traverse
   C) They cannot be implemented in C
   D) They are always sparse
   Answer: B

10. If an array A[20] starts at address 2000, and each integer occupies 4 bytes, what is the address of A[10]?
    A) 2040
    B) 2044
    C) 2048
    D) 2046
    Answer: B

---

11. Which of these is an advantage of arrays?
    A) Dynamic resizing
    B) Random access in O(1)
    C) Easy insertion at random index
    D) Efficient deletion
    Answer: B

12. Which algorithm uses arrays for its core implementation?
    A) Merge sort
    B) Hash tables
    C) Quick sort
    D) All of the above
    Answer: D

13. The largest element in an array can be found in:
    A) O(1)
    B) O(n)
    C) O(log n)
    D) $O(n^2)$
    Answer: B

14. Which of the following traversals is used to compute prefix sums efficiently in an array?
    A) Forward traversal
    B) Backward traversal
    C) Both forward and backward
    D) None
    Answer: A

15. Which of these is not a valid application of arrays?
    A) Implementing a matrix
    B) Polynomial representation
    C) Stack implementation
    D) Hash table with chaining
    Answer: D

16. Which of these will result in an out-of-bound error?
    A) Accessing index -1
    B) Accessing index = size of array
    C) Accessing index > size of array
    D) All of the above
    Answer: D

17. Which of these is true about sparse arrays?
    A) Most of the elements are non-zero
    B) Most of the elements are zero
    C) They cannot be stored in arrays
    D) They always need 2D arrays
    Answer: B

18. Which is more cache-friendly?
    A) Array
    B) Linked List
    C) Stack
    D) Queue
    Answer: A

19. Inserting an element in the middle of an array takes:
    A) O(1)
    B) O(log n)
    C) O(n)
    D) $O(n^2)$
    Answer: C

20. Which of these sorting algorithms is best when the input is stored in an array and cache performance is important?
    A) Merge sort
    B) Quick sort
    C) Bubble sort
    D) Selection sort
    Answer: B

21. The address of element A[i][j] in a 2D array stored in row-major order is:
    A) Base + (i * row_size + j) * w
    B) Base + (j * row_size + i) * w
    C) Base + i + j
    D) Base + (i+j)*w
    Answer: A

22. Which is better for representing polynomial coefficients with large degrees?
    A) Linked List
    B) Arrays
    C) Stacks
    D) Queues
    Answer: B

23. What is the time complexity to reverse an array in place?
    A) O(1)
    B) O(log n)
    C) O(n)
    D) O(n log n)
    Answer: C

24. Which of these allows efficient searching in O(log n) with arrays?
    A) Sorted array
    B) Unsorted array
    C) Sparse array
    D) Randomized array
    Answer: A

25. Which of these is not an in-place array operation?
    A) Reversing
    B) Swapping
    C) Sorting
    D) Copying
    Answer: D

26. Which of these is true about dynamic arrays (like Python lists)?
    A) They double size when capacity is reached
    B) They cannot shrink
    C) They never reallocate memory
    D) They always allocate infinite memory
    Answer: A

27. In a fixed-size array, what happens if you try to insert beyond capacity?
    A) Overwrites existing element
    B) Throws overflow error
    C) Resizes automatically
    D) Appends NULL
    Answer: B

28. The best sorting algorithm for arrays with small size is:
    A) Bubble sort

B) Selection sort
C) Insertion sort
D) Quick sort
Answer: C

29. Which of these can be used for implementing circular queues?
A) Arrays
B) Linked lists
C) Both A and B
D) Neither
Answer: C

30. Which of these is a disadvantage of arrays?
A) Cache inefficiency
B) O(1) indexing
C) Easy traversal
D) Predictable memory layout
Answer: A

---

31. Which of the following is an application of arrays in CPU scheduling?
A) Round-robin
B) Priority scheduling
C) Both A and B
D) None
Answer: C

32. Which of these is used to find the majority element in an array efficiently?
A) Boyer-Moore algorithm
B) Quick sort
C) Binary search
D) Merge sort
Answer: A

33. Which of these uses prefix arrays?
A) Range sum queries
B) Hashing
C) Graph traversal
D) String matching
Answer: A

34. Which is the correct method for rotating an array by k elements?
A) Using extra array of size k
B) Using reversal algorithm
C) Using temporary variable
D) All of the above
Answer: D

35. What is the complexity of merging two sorted arrays of size m and n?
A) O(m+n)

B) O(log(m+n))
C) O(mn)
D) O(n²)
Answer: A

36. Which of these requires shifting elements in an array?
A) Traversal
B) Random access
C) Insertion at beginning
D) Accessing last element
Answer: C

37. The advantage of row-major over column-major storage is:
A) Cache locality in row traversal
B) Faster column access
C) Both are same
D) Saves memory
Answer: A

38. What is the time complexity of finding the second largest element in an unsorted array?
A) O(1)
B) O(n)
C) O(log n)
D) O(n²)
Answer: B

39. Which of these is true about sparse matrices stored as arrays?
A) They waste memory
B) They save memory
C) They cannot be stored in arrays
D) They are faster
Answer: A

40. Which is the best approach to find duplicates in an unsorted array?
A) Sorting + scan
B) Hashing
C) Nested loops
D) All of the above
Answer: D

---

41. Which of these algorithms uses arrays for partitioning?
A) Merge sort
B) Quick sort
C) Heap sort
D) BFS
Answer: B

42. Which data structure is used to implement heaps?
A) Linked list

B) Array
C) Queue
D) Tree only
Answer: B

43. What is the worst-case time for searching in a sorted array using binary search?
A) O(1)
B) O(n)
C) O(log n)
D) O(n log n)
Answer: C

44. Which of these is not true about arrays?
A) They allow direct memory access
B) They always consume less memory than linked lists
C) They are contiguous
D) They allow O(1) access
Answer: B

45. Which algorithm can find the equilibrium index of an array efficiently?
A) Prefix sum method
B) Binary search
C) Hashing
D) Greedy
Answer: A

46. What is the auxiliary space complexity of merge sort on arrays?
A) O(1)
B) O(log n)
C) O(n)
D) O(n^2)
Answer: C

47. Which technique is efficient for finding missing numbers in a range?
A) Sorting
B) Hashing
C) XOR operation
D) All of the above
Answer: D

48. Which of these allows two stacks to share one array efficiently?
A) Push-pop method
B) Two pointers growing towards each other
C) Circular method
D) Resizing method
Answer: B

49. Which is the best way to represent a matrix with mostly zero elements?
A) 2D array
B) Sparse representation using arrays
C) Linked list

D) Tree
Answer: B

50. Which of these is true about arrays vs linked lists?
A) Arrays allow O(1) random access, linked lists don’t
B) Linked lists are always more memory efficient
C) Arrays always resize automatically
D) Linked lists are cache-friendly
Answer: A