1. What is the minimum number of comparisons required to find the largest element in an array of N elements?

Ans. N-1

2. What is the minimum number of comparisons to find the 2nd largest element provided you have found the largest element.

Ans. logN-1

3. In Array Data Structure, what is the biggest limitation?

Ans. Pre-defined array size

4. In Dynamic Array, what is the worst case Time Complexity of inserting an new element?

Ans. O(n)

5. If we have N elements, what is the maximum size of the Dynamic Array?

Ans. (N left shift 1) AND (NOT N)

6. What is the average case Time Complexity to delete a specific element in an array?

Ans. O(n)

7. How is an N-dimensional array stored in memory?

Ans. 1D array

8. To move all negative numbers to the front of array, how many traversals are needed?

Ans. 1

9. Array was improved by Dynamic Array. Dynamic Array is improved by?

Ans. Hashed Array Tree

10. What is the Time Complexity to create Suffix Array?

Ans. O(N logN logN)

11. Using Rolling Hash technique in array, the time complexity to find the hash of a sub-array is?

Ans. O(1)

12. Which data structure is used to find the least frequent element in an array?

Ans. Hash Map

13. Which Algorithm is used to find the largest sub-array sum in optimal time O(N)?

Ans. Kadane's algorithm

14. Boyer Moore voting algorithm is used to find the majority element among the given sequence of elements in an array which occurs more than N/2 times. What is the Time Complexity provided space complexity is O(1)?

Ans. O(N)

15. If there are 1024 elements and we have found the largest element, how many comparisons are needed to find the 2nd largest element?

Ans. 9 (If there are 1024 elements and we have found the largest element, we need only 9 comparisons to find the 2nd largest element. 9 = log(1024) - 1. The minimum number of comparisons required to find the 2nd largest element is N + logN - 2. Of this, N-1 comparisons are required to find the largest element. So, only logN - 1 extra comparisons are needed to find the 2nd largest element. )

16. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is (GATE CS 2002)

Ans. N

17. Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest? (GATE CS 2004)

Ans. union, intersection

18. Which of the following points is/are true about Linked List data structure when it is compared with array

Ans. A. Arrays have better cache locality that can make them better in terms of performance.

B. It is easy to insert and delete elements in Linked List

C. Random access is not allowed in a typical implementation of Linked Lists

D. The size of array has to be pre-decided, linked lists can change their size any time.

19. What is the time complexity to count the number of elements in the linked list?

Ans. O(n)

20. What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?

Ans. θ (n)

21. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

Ans. n

22. Which of these is an application of linked lists?

Ans. A. To implement file systems

B. For separate chaining in hash-tables

C. To implement non-binary trees

23. In circular linked list, insertion of node requires modification of?

Ans. Two Pointer

24. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only.

Given the representation, which of the following operation can be implemented in O(1) time?

Ans. Insertion at the end of the linked list

Deletion of the front node of the linked list

25. What would be the asymptotic time complexity to find an element in the linked list?

Ans. O(n)

26. The concatenation of two list can performed in O(1) time. Which of the following variation of linked list can be used?

Ans. Circular doubly linked list

27. Consider the following definition in c programming language.Which of the following c code is used to create new node?

Ans. ptr = (NODE*)malloc(sizeof(NODE));

28. What kind of linked list is best to answer question like "What is the item at position n"?

Ans. Array implementation of linked list

29. Linked lists are not suitable to for the implementation of?

Ans. Binary search

30. Linked list is considered as an example of _____ type of memory allocation.

Ans. Dynamic

31. In Linked List implementation, a node carries information regarding

Ans. Link

32. Linked list data structure offers considerable saving in

Ans. Space Utilization and Computational Time

33. Which of the following points is/are true about Linked List data structure when it is compared with array

Ans. A. Arrays have better cache locality that can make them better in terms of performance

B. It is easy to insert and delete elements in Linked List

C. Random access is not allowed in a typical implementation of Linked Lists

34. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

Ans. Merge Sort

35. Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?

Ans. Possible if X is not last node

Explanation: Following are simple steps.

```
struct node *temp = X->next;
X->data = temp->data;
X->next = temp->next;
free(temp);
```

36. The time complexity of the binary search algorithm is?

Ans. O(log n)

37. In the iterative method, the space complexity would be?

Ans. O(1)

37. In the recursive method, the space complexity would be?

Ans. O(log n)

38. The complexity of Binary search algorithm is

Ans. O(log n)

39. The complexity of merge sort algorithm is

Ans. O(n log n)

40. The complexity of Bubble sort algorithm is

Ans. O(n2)

41. The worst case complexity for insertion sort is

Ans. O(n^2) Auxiliary Space: O(1)

42. Auxiliary Space for insertion sort is

Ans. O(1)

43. The worst case complexity of quick sort is

Ans. O(n^2)