Q1. Which one of the following is not a component of data structure?

A. Operations
B. Storage structures
C. Algorithms
D. Programming Language

Q2. Which of the following points are considered while selecting a data structure for an application?

A. Resource constraints
B. Time and Space complexity of operations
C. Programming language
D. Frequent operations to be performed

Q3. What is the value of $\sum_{i=1}^{n}(2i-1) = ?$

A. n
B. n^2
C. n^3
D. n^2-n

Q4 The following statement is true or false

Nlog N and Nlog (N^2) grow at the same rate.

A. True
B. False

Q5. If an algorithm has the time complexity θ(nLogn), select the following options which are also correct

A. BigO(nLogn)
B. Smallo (nLogn)
C. BigOmega(nLogn)
D. SmallOmega (nLogn)

Q6. If f(n) = Θ(g(n)) & g(n) = BigO(h(n)) Then

A. f(n) = Θ(h(n))
B. f(n) = BigO(h(n))
C. f(n) = BigOmega(h(n))
D. f(n) = SmallOmega(h(n))

Q7. Suppose T1(n) = O(f(n)) and T2(n) = O(f(n)) then select the correct statements?

A. T1(n) + T2(n) = O(f(n))
B. T1(n) - T2(n) = O(f(n))
C. T1(n) * T2(n) = O(f(n))
D. T1(n) / T2(n) = O(f(n))

Q8. Which is/are true (for any constant c , k):

A. c is O(log *n*) but log *n* is not O(1)
B. $c * n^k$ is $O(n^k)$
C. $c^2 * n^2$ is $O(n^2)$
D. $c * n * \log_k n$ is $O(\log_k n)$

Q9. An algorithm should have _____ well-defined output

A. 0
B. 0 or more
C. 1 or more
D. Unknown

Q10. The running time complexity of a linear time algorithm is given as

A. Θ (1)
B. Θ (n)
C. Θ (n log n)
D. Θ (n^2)

Q11.The time complexity of the following program will be

```
sum = 0;
for( i=0; i<n; i++ ){
        for( j=0; j<n*n; j++ )
                sum++;
}
```

A. O(N)
B. O(NlogN)
C. $O(N^2)$
D. $O(N^3)$

Q12. The following summation of three notations $\mathbf{O}(n^2) + \Theta(n^2) + \Omega(n^2)$ can be written as

A. $O(n^2)$
B. $\Theta(n^2)$
C. $\Omega(n^2)$
D. $\Theta(n^3)$

Q13. Select the correct options for the following recursive equation: T(n)=T(n-1)+10n-100Logn

A. Θ (nLogn)
B. $\Theta(n^2)$
C. Θ (n)
D. $\Theta(n^3)$

Q14 Programs A and B are analyzed and found to have worst-case running times no greater than 150nlog(2n) and n^2, respectively. Which program has the better guarantee on the running time, for large values of n (n > 10,000)?

A. Program A
B. Program B
C. Information is not sufficient
D. Both will be same

Q15 Programs A and B are analyzed and found to have worst-case running times no greater than 150nlog(2n) and n^2, respectively. Which program has the better guarantee on the running time, for small values of n (n < 100)?

A. Program A
B. Program B
C. Information is not sufficient
D. Both will be same

Q16 Programs A and B are analyzed and found to have worst-case running times no greater than 150nlog(2n) and n^2, respectively. Is it possible that program B will run faster than program A on all possible inputs?

A. Yes
B. No
C. Information is not sufficient

Q17. Master Theorem is used to solve

A. Recurrence functions
B. Iterative functions
C. Dynamic Functions
D. none of these

Q18. When new data are to be inserted into a data structure, but there is no available space, this situation is usually called

A. Houseful
B. Underflow
C. Overflow
D. Maximum

Q19. In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into a NONEMPTY queue?

A. Only front pointer
B. Only rear pointer
C. Both front and rear pointer
D. No pointer will be changed

Q20. Suppose we have a list of elements [0..n -1] and we want to delete all duplicates. last_position is initially n - 1, but gets smaller as elements are deleted. Consider the pseudocode program fragment as shown bellows. The procedure DELETE deletes the element in position j and collapses the list.

```
for( i=0; i<last_position; i++ )
{
        j = i + 1;
        while( j<last_position )
                if( a[i] == a[j]
                        DELETE(j);
                else
                        j++;
}
```

What is the running time of the algorithm while list of elements is implemented using a linked list?

A. Θ (nLogn)
B. Θ (n^2)
C. Θ (n)
D. Θ (n^3)

Q21. If the address of A[1][1] and A[2][1] are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in _____ order

A. row major
B. column major
C. matrix major
D. none of above

Q22. The base address of a 2d array *A[0][0]* of size (4x4) is 1000 and *w* is 4bytes. The address of the *A*[2][3] will be, if row major order is selected

A. 1000
B. 1036
C. 1040
D. 1044

Q23. A program P reads in 1000 integers in the range [0, 200] representing the scores of 1000 students. It then prints the frequency of each score below 50. What would be the best way for P to store the frequencies?

A. An array of 50 numbers
B. An array of 100 numbers
C. An array of 500 numbers
D. An array of 200 numbers

Q24. Which of the following is the way to represent Sparse Matrix?

A. Array
B. Linked list
C. Multi Linked List
D. None of the above

Q25. The time complexity to perform the transpose of a sparse matrix will be? The matrix is represented as triplet and contains n rows, n column and 2n non-zero elements?

A. θ(Logn)
B. θ(nLogn)
C. θ(n^2)
D. θ(n)

Q26. Following are the two sparse matrices represented as triplet
A= (1,1,5),(1,2,10),(1,3,15),(2,2,2),(2,3,1)
B=(1,1,3),(2,1,4),(3,2,6)
Write down the multiplication result in the triplet form

Q27. What is the output of following function for *start* pointing to first node of following linked list? 1->2->3->4->5->6

```
void fun(struct node* start)
{
        if(start == NULL)
                return;
        printf("%d ", start->data);
        if(start->next != NULL )
                fun(start->next->next);
        printf("%d ", start->data);
}
```

A. 1 4 6 6 4 1
B. 1 3 5 1 3 5
C. 1 2 3 5
D. 1 3 5 5 3 1

Q28. Which of the following instructions would create the second node in the list? Assume the first node has been created and is referenced by head.

A. head.next = new_Node;
B. head.next.next = new_Node;
C. head = new_Node;
D. head.next.next.next = new_Node;

Q29. 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 θ (n) time?

A. Insertion at the front of the linked list
B. Insertion at the end of the linked list
C. Deletion of the front node of the linked list
D. Deletion of the last node of the linked list

Q30. 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?

A. Insertion at the front of the linked list
B. Insertion at the end of the linked list
C. Deletion of the front node of the linked list
D. Deletion of the last node of the linked list

Q31. Which of the following is/ are TRUE?

A. A circularly linked list is a linked list in which the node at the tail of the list points to NULL
B. A circularly linked list is a linked list in which the node at the tail of the list points back to the node at the head of the list.
C. When we add an item to a queue, we say we push it onto the queue.
D. When we add an item to a stack, we say we push it onto the stack.

Q32. For which of the following operations is a doubly linked list better than a single linked list:

A. Traversing the list to process each node
B. Deleting a node whose location LOC is already assigned to pointer PTR
C. Searching an unsorted list for a given element ITEM
D. Inserting a node before the node with a given location LOC

Q33. In linked list each node contain minimum of two fields. One field is data field to store the data second field is?
A. Pointer to character

B. Pointer to integer
C. Pointer to node
D. Node

Q34. Linked list is considered as an example of _____ type of memory allocation.
A. Dynamic
B. Static
C. Compile time
D. None of the mentioned

Q35. Which is the pointer associated with the availability list?

A. FIRST
B. AVAIL
C. TOP
D. REAR

Q36. In linked list implementation of queue, if only front pointer is maintained, which of the following operation would be efficient?

a) Insertion
b) Deletion
c) Searching
d) Sorting

Q37. While converting infix notation to postfix notation

A. Operand is always placed in the output
B. Higher and equal priority operators are considered same
C. Parenthesis are included in the output
D. Operator is placed in the stack when the stack operator has higher precedence

Q38. Which of the following is an application of stack?

A. Finding factorial
B. Tower of Hanoi
C. Infix to postfix conversion
D. BFS traversing of a tree

Q39.The postfix expression for the infix expression: A + B* (C+D) /F + D*E is

A. AB+CD+*F/D+E*
B. ABCD+*F/+DE*+
C. A*B+CD/F*DE++
D. A+*BCD/F*DE++

Q40. Write the prefix form of the following expression (write answer without any space or comma eg. +*ABC)
(A + B) * C + D / (E + F * G) – H

Q41. Write the prefix form of the following expression (write answer without any space or comma eg. +*ABC)
A+((B - C * D)/E)+F-G/H

Q42. What is the value of following postfix expression
54 6 + 7 4 - * 9 / 35 15 + +

Q43. While converting the following infix expression to postfix expression, what is the maximum number of entries (operand/operator/bracket) will be there in the stack at a time:
(A * B - (C - D)) / (E + F)

A. 2
B. 3
C. 4
D. 5

Q44. The time and space complexity to evaluate a prefix notation is?

A. θ(1), O(1)

B. θ(1), O(n)
C. θ(n), θ(n)
D. θ(n), O(n)

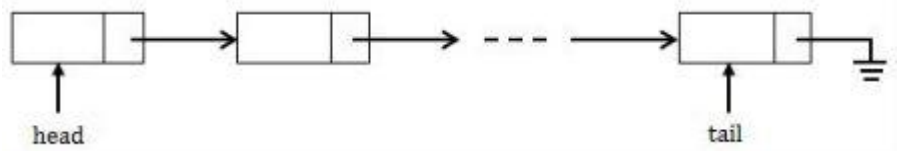
Q45. A parentheses checker program would be best implemented using
A. List
B. Queue
C. Stack
D. Any of the above

Q46. How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you.
A. 1
B. 2
C. 3
D. 4

Q47. A priority queue can be efficiently implemented using which of the following data structures? Assume that the number of insert and peek (operation to see the current highest priority item) and extraction (remove the highest priority item) operations are almost same
A. Array
B. Linked List
C. Heap Data Structures
D. None of the above

Q48. Select the correct statements:
A. First-in-first out types of computations are efficiently supported by STACKS.
B. Implementing LISTS on linked lists is more time efficient than implementing LISTS on an array for search operation.
C. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
D. Last-in-first-out type of computations is efficiently supported by QUEUES.

Q49. A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let n denote the number of nodes in the queue. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail.



Which one of the following is the time complexity of the most time-efficient implementation of 'enqueue' and 'dequeue, respectively, for this data structure?
A. θ(1), θ(1)
B. θ(1), θ(n)
C. θ(n), θ(1)
D. θ(n), θ(n)

Q50. A Circular Queue of size four stores int values. Initially the queue is empty, and Front and Rear both are equal to -1. What will be the value of Front and Rear after the following instructions are executed.
enqueue(6); enqueue(12); enqueue(13); dequeue( ); dequeue( ); enqueue(19); enqueue(21); enqueue(22); dequeue( ); enqueue(20);
A. Front=2, Rear=3
B. Front=3, Rear=2
C. Front=1, Rear=
D. Front=2, Rear=1

Q51. If the two lists to be concatenate, which of the following implementations of a list will be most efficient in term of time complexity?

A. singly linked list
B. doubly linked list
C. circular doubly linked list
D. array implementation of lists

Q52. What is the time complexity to count the number of elements in the linked list?

A. O(1)
B. O(n)
C. O(logn)
D. None of the mentioned

Q53. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

A. log 2 n
B. n/2
C. log 2 n – 1
D. n

54Select the correct statements:

A. The number of comparisons for sequential search is O(n)
B. The binary search algorithm is BigOmega(n)
C. The binary search algorithm is O(log n)
D. The binary search algorithm is Theta(n)

Q55. Select the correct options

A. Time complexity of Binary search will be asymptotically better than Ternary Search
B. Ternary search is more time complex as compared to binary search
C. Space complexity of the Binary search when implemented in iterative or recursive approach is same
D. Ternary search will be better if the desired element is at the beginning

Q56. The best case time and space complexity of the Binary search (recursive approach) is

A. O(1), O(1)
B. O(logn),O(1)
C. O(1),O(logn)
D. O(logn),O(logn)

Q57. The best case time and space complexity of the Exponential search which uses Binary search (recursive approach) is ?

A. O(1), O(1)
B. O(logn),O(1)
C. O(1),O(logn)
D. O(logn),O(logn)

Q58. The worst case time and space complexity of the Exponential search which uses Binary search (recursive approach) is ?

A. O(n), O(n)
B. O(logn),O(1)
C. O(1),O(logn)
D. O(logn),O(logn)

Q59. Select the correct statements

A. Exponential search performs better than the binary search in all cases
B. Exponential search performs better than the binary search if the desired item is at the beginning
C. Exponential search performs better than the binary search if the desired item is at the middle
D. Exponential search performs better than the binary search if the desired item is at the end

Q60. In the worst case, the number of comparisons needed to search a sorted array of length n for a given element is

A. log(2n)
B. n/2
C. log(n)+1
D. n