

UNIT – II

1. What are all the Common data structures?

- Stacks
- Queues
- Lists
- Trees
- Graphs
- Tables

2. Define Stack and its basic operations of stack?

It is a collection of items where new items may be *inserted* or *deleted* only at one end, called the **top** of the stack. A stack is a data structure that keeps objects in Last- In-First-Out (**LIFO**) order, Objects are added to the top of the stack. Only the top of the stack can be accessed. The various operations are

Push: Push is performed by inserting at the top of stack.

Pop: pop deletes the most recently inserted element.

Top: top operation examines the element at the top of the stack and returns its value.

3. Define abstract data type(ADT) and list its advantages?

ADT is a set of operations. An abstract data type is a data declaration packaged together with the operations that are meaning full on the data type.

Abstract Data Type

- Declaration of data
- Declaration of operations.

Advantages:

- Easy to debug small routines than large ones.
- Easy for several people to work on a modular program simultaneously.
- A modular program places certain dependencies in only one routine, making changes easier.

4. What are the operations of ADT?

Union, Intersection, size, complement and find are the various operations of ADT.

5. State the different ways of representing expressions?

The different ways of representing expressions are

- Infix Notation
- Prefix Notation
- Postfix Notation

6. State the rules to be followed during infix to postfix conversions?

- Fully parenthesize the expression starting from left to right. During parenthesizing, the operators having higher precedence are first parenthesized
- Move the operators one by one to their right, such that each operator replaces their corresponding right parenthesis
- The part of the expression, which has been converted into postfix is to be treated as single operand

7. Write the various applications of the stack?

- Towers of Hanoi
- Reversing a string
- Balanced parenthesis
- Recursion using stack
- Evaluation of arithmetic expressions
- Infix to postfix conversion

8. Write the algorithm for balancing symbols?

1. Make an empty stack.
2. Read characters until end of file.
3. If the character is an opening symbol, then push it onto the stack.
4. If it is a closing symbol
Then If the stack is
empty Report an error
Otherwise pop the
stack
5. If the symbol popped is not the corresponding opening
symbol Then
Report an error
6. If the stack is not empty at the end of
file Then
Report an error.

9. Define Infix notation?

- ☐ The operator symbol is placed in between its two operands. This is called ***infix notation***.

*Example: $A + B$, $E * F$*

- ☐ Parentheses can be used to group the operations.

*Example: $(A + B) * C$*

10. Define Prefix notation?

- ☐ Polish notation refers to the notation in which the operator symbol is placed before its two operands. This is called **prefix notation**.

*Example: +AB, *EF*

11. Define Postfix notation?

- ☐ **Reverse Polish Notation** refers to the analogous notation in which the operator symbol is placed after its two operands. This is called **postfix notation**.

*Example: AB+, EF**

- ☐ Here also the parentheses are not needed to determine the order of the operations.

12. Convert the given infix to postfix and prefix? $(j*k)+(x+y)$

postfix : $jk* xy++$

prefix : $+*jk+xy$

13. Convert into postfix and evaluate the following expression.? $(a+b*c)/d$; $a=2$

$b=4$ $c=6$ $d=2$

Post fix:

$abc*+d/$

Evaluation:

$2\ 4\ 6\ * \ +2/ = 13$

14. Write the features of representing calls in a stack?

- When a function is called the register values and return address are saved.
- After a function has been executed the register values are resumed on returning to the calling statement.
- The stack is used for resuming the register values and for returning to the calling statement.
- The stack overflow leads to fatal error causing loss of program and data.
- Recursive call at the last line of the program is called tail recursion and also
- leads to error.

15. What is the data structures used to perform recursion?

Stack. Because of its LIFO (Last In First Out) property it remembers its 'caller' so knows whom to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function calls.

Every recursive function has its equivalent iterative function. Even when such equivalent iterative procedures are written, explicit stack is to be used.

16. Define a Queue?

Queue is an ordered collection of elements in which insertions are restricted to one end called the rear end and deletions are restricted to other end called the front end. Queues are also referred as First-In-FirstOut (FIFO) Lists.

17. List the operations of queue?

Two operations

- Enqueue-inserts an element at the end of the list called the rear.
- Dequeue-deletes and returns the element at the start of the list called as the front.

18. Write the routines for enqueue operation in queue?

```
Void Enqueue(Element type X,Queue Q)
{
If(IsFull(Q)) Error("Full queue"); Else
{ Q->Size++;
  Q->Rear=Succ(Q->Rear,Q); Q->Array[Q->Rear]=X; } }
```

19. Write the routines for dequeue operation in queue?

```
Void Dequeue(Queue Q)
{
If(IsEmpty(Q)) Error("Empty Queue"); Else
Q->front++;
}
```

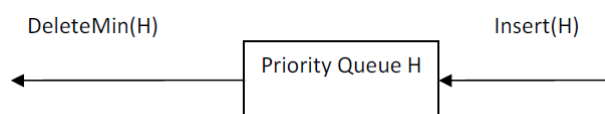
20. What is the difference between a queue and a stack?

Queue	Stack
Queue is typically FIFO	Stack is LIFO
Elements get inserted at one end of a queue and retrieved from the other	Insertion and removal operations for a stack are done at the same end.

21. What are priority queues?

A priority queue is a collection of elements such that each element has been assigned a priority.

- Insert-inserts an element at the end of the list called the rear.
- DeleteMin-Finds, returns and removes the minimum element in the priority Queue.



22. What are the types of priority queues?

- Ascending Priority Queue
- Descending Priority Queue

29. What is an ascending priority queue?

It is a collection of items into which items can be inserted arbitrarily and from which the smallest item can be removed.

30. What is a descending priority queue?

It is a collection of items into which items can be inserted arbitrarily and from which the largest item can be removed.

31. Give the applications of priority queues?

There are three applications of priority queues

1. External sorting.
2. Greedy algorithm implementation.
3. Discrete event simulation.
4. Operating systems.

33. Define a Dequeue?

Deque is a list that combines the properties of a stack and queue. It is a linear list in which insertions and deletions are made to or from either end of the structure.

34. List the applications of Queues?

- Checking strings of a language
- Queuing theory simulation
- Input / Output buffers
- Graph searching
- Jobs sent to a printer
- Line in a ticket counter

35. List down the rules to evaluate the postfix expression.

- Initialize an empty stack
- While token remains in the input stream
 - Read next token
 - If token is a number, push it into the stack
 - Else, if token is an operator, pop top two tokens off the stack, apply the operator, and push the answer back into the stack
- Pop the answer off the stack

36. Define recursive function

A Function which calls itself known as recursive function.

Eg.

```
Void main()  
{  
    printf(" Example recursive function with infinite  
    loop"); main();  
}
```

37. Differentiate recursion and iteration

Iteration :-

- Keep repeating the steps until the task is completed.
- **Eg.** loop counter reaches its maximum value

Recursion:-

- Breaking the problem into smaller pieces to solve it and combining the results.
- Using recursive method in programming is elegant but not efficient always.
- **Eg.** Finding factorial of a number using recursive call.

38. Explain the mechanism of queue shortly

Queue uses FIFO (First In First Out) strategy. i.e. First element added to the queue would be the first element to be removed. Insertion and deletion operations are carried out at REAR and FRONT ends respectively. Adding an element to the queue is known as enqueue process and removing an element from queue is dequeue process.

39. What are the various operations that can be performed on Queue?

In Queue, we can perform two operations namely Insertion and Deletion.

Insertion: Insertion can be done by incrementing Rear by 1.

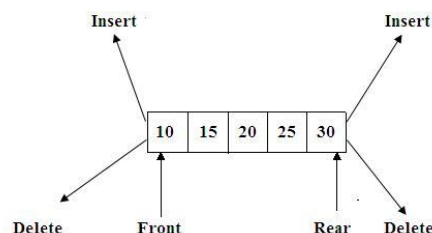
Deletion: Deletion can be done by incrementing Front by 1

40. What do you mean by Queue empty?

- If Queue has no data, then it is called as Queue Empty.
- Queue may be empty on two conditions.
 - i. Front==Rear== -1 and
 - ii. Front==Rear.

41. Define Dequeue?

- Dequeue is otherwise called as Double ended queue.
- In Dequeue, we can insert and delete at both ends either front or rear.



42. What are the types of Queue?

- Simple Queue
- Circular Queue
- Priority Queue
- Double ended Queue

43. What is difference between Queue and Dequeue?

- A queue is designed to have elements inserted at the end of the queue, and elements removed from the beginning of the queue.
- Whereas Dequeue represents a queue where you can insert and remove elements from both ends of the queue.

44. What is Priority Queue with example?

- A priority queue is a special type of queue in which each element is associated with a priority and is served according to its priority.
- Generally, the value of the element itself is considered for assigning the priority.
- For example: The element with the highest value is considered as the highest priority element.

45. Define Ascending Priority Queue.

- In this elements are placed in ascending order.
- The first smallest element is placed in first position and second smallest element in second position and so on.
- The new data item is inserted in priority queue without affecting the ascending order of queue.

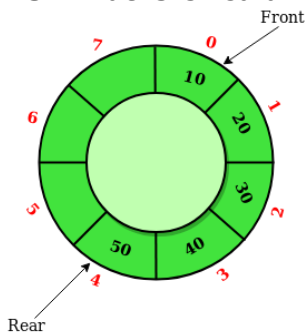
46. Define Descending Priority Queue.

- In this elements are placed in descending order.
- The first highest element is placed in first position and second highest element is placed in second position and so on.
- The new data item is inserted in priority queue without affecting the descending order of queue.

47. What are the application of Priority Queue?

- Priority queues are used to sort heaps.
- Priority queues are used in operating system for load balancing and interrupt handling.
- Priority queues are used in huffman codes for data compression.
- In traffic light, depending upon the traffic, the colors will be given priority.

48. What is Circular Queue?



- Circular Queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle.
- The last position is connected back to the first position to make a circle.
- Circular Queue is used in memory management and scheduling process.
- It is also called 'Ring Buffer'.

49. How do you determine the size of Circular Queue?

Assuming you are using array of size N for queue implementation, then size of queue would be **size = (N-front + rear) mod N**. This formula work for both liner and circular queues.

50. Write the limitations of stack.

- Only a small number of operations can be performed on it.
- It contains only a bounded capacity

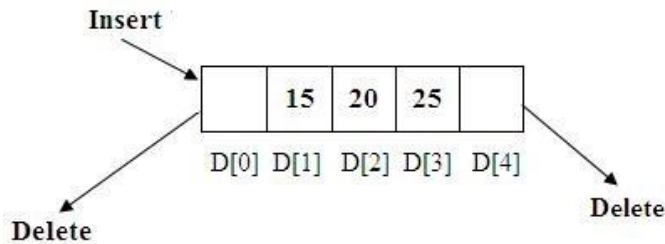
51. What are the types of Dequeue?

Two types of Dequeue are

1. Input Restricted Dequeue
2. Output Restricted Dequeue.

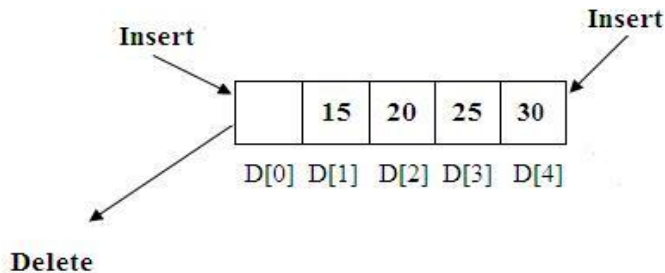
52. Define Input restricted Dequeue.

- It means we can insert the element only at one end and delete the elements at both ends.
- The above diagram shows the input restricted Dequeue.



53. Define Output restricted Dequeue.

- It means we can insert the elements in both ends and delete the elements at only one end.
- The above diagram shows the output restricted Dequeue.



54. State the advantages of using infix notations.

The advantages of using infix notations are,

- It is the mathematical way of representing the expression.
- It is easier to see visually which operation is done from first to last.

55. State the advantages of using postfix notations.

The advantages of using postfix notations are

- We need not worry about the rules of precedence.
- We need to worry about the rules for right to left associatively.
- We need not parenthesis to override the above rules.

56. Write down the application of queue.

- Printing
- CPU scheduling
- Mail service
- Elevator
- Keyboard buffering