Unit – II

Chapter -9 & 10
ARRAYS & LINKED LIST

Defination of Datastructure

A **data structure** is a logical method of representing data in memory using the simple and complex data types provided by the language.

Arrays:

An array is collection of the homogeneous elements that are referred by a common name. It is also called a subscripted variable as the elements of an array are used by the name of an array and an index or subscript.

Array are of two types:

1. One-dimensional arrays 2. Multi-dimensional arrays.

Address Calculation:

The array elements are stored in contiguous memory locations by sequential allocation technique.

Address of arr[i] = B+(I-LB)*S

Sequential Allocation:

The process of storing elements in a fixed order in a data structure where the time required for such access is dependent on the order of the elements.

Using Row Major order the add of a [i] [j] is given by,

Address of a [i] [j]= B+[(i-LB1)*N+(j-LB2)]*S

Using column Major order the add of a [i] [j] is given by,

Address of a [i] [j]= B+[(i-LB1)+(j-LB2)*M]*

One Dimensional Array:

The simplest type of data structure is a one dimensional array in which each elements of a linear array is referenced by one subscript.

The operations one normally perform on any linear structure include the following:

(a) **Traversal**: processing each elements in the list.

(b) **Search**: finding the location of the element with a given value or the record with a given key.

Linear Search: this is a neutral searching method in which we search for a element by traversing through the entire list from beginning until we get the element.

Binary Search: the search can be made faster by an algorithm in which the list or the array should be sorted in advance is called binary search.

(c) **Insertion**: adding a new element to the list.

(d) **Deletion**: removing an element from the list.

(e) **Sorting:** arranging the elements in some types of order.

Insertion sort: this sorting method divides the list into sorted part and unsorted part. It picks up one element from the front of the unsorted part and inserts it at its proper place in the sorted part and repeats this action till the unsorted part is exhausted.

Selection sort: it is very simple way of sorting data stored in a one dimensional array. All the elements in the array are searched for the smallest element. The selected element is exchanged with the first element in the array.

Bubble sort: it is also very simple sorting algorithm. This method proceeds by looking the array for left to right, and whenever a pair of adjacent elements is found to be out of order, the elements are exchanged. Therefore, after the first pass, the largest element is the array will have bubble up to one end of the array.

(f) **Merging**: merging is the process of combining two or more sorted arrays into another array which is also sorted.

Concatenation of Two Linear Array

***Concatenation means joining the element** of two arrays to form a new array*. First copy all the elements of one array and then copy all the elements of other array into the new array. The size of the concatenated array must be equal to or greater than that of the sum of sizes of the two arrays to be concatenated.

Two Dimensional Array:

A two dimensional array(having two subscripts) is suitable for table processing or matrix manipulation.

Traversal: it means visiting each element one after the other.

Finding sum/difference of two N×M arrays containing numeric values:

The sum/difference of two N×M arrays containing numeric values can be obtained by adding/subtracting the corresponding elements of the two arrays. The result can either be directly displayed on monitor or stored into third array of size N×M and then this resulting array can be displayed.

For example, the sum of two arrays A and B of size 3×2 is shown below.

4	6	5	2	9	8
7	1	1	8	8	9
5	3	2	4	7	7
Array A		Array B		resultant after addition	

If difference of two arrays of size 3×3 is required then we get result as illustrated by the following

example :

40	25	73	31	20	41	9	5	32
18	55	29	10	36	15	8	19	14
70	62	47	20	32	17	50	30	30
Array A			Array B			Result after difference		

We may even have negative values as array elements or after sum/difference.

Interchanging Row and Column elements in a Two Dimensional Array:

Interchanging means swapping the values. The elements of row and column can be interchanged only if the array is of size N×N as the result is to be stored in the array itself.

For example,

1	2	3	1	4	7
4	5	6	2	5	8
7	8	9	3	6	9
Given	Array	Array after interchange			

Stacks :

A stack is a list of elements in which an element may be inserted or deleted only at one end, called the TOP of the stack. This means, in particular, that elements are removed from a stacks in the reverse order of that in which they were inserted into the stack.

A stack is a **last-in first-out** or ***LIFO*** data structure.

(a) "Push" s the term used to insert an element into a stack.
(b) "Pop" is the term used to delete an element from the stack.

Queue:

A queue is a linear list of elements in which deletions can take place only at one end, called the front, and insertions can take place only at the other end, called the rear. The terms "front" and "rear" are used in describing a linear list only when it is implemented as a queue.
A queues are also called **first-in first-out(FIFO)** list.

Linked List:

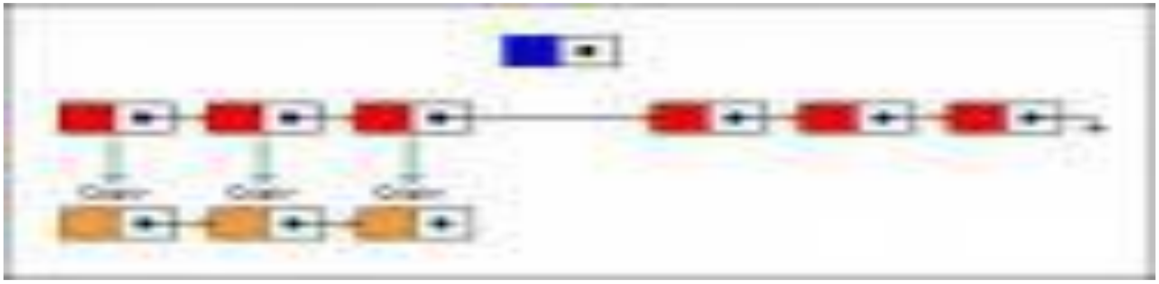
A linked list is a linear collection of data elements, called nodes pointing to the next nodes by means of pointers.

Stack As A linked List

A linked list is a dynamic data structure where space requirements need not be predetermined. A stack implemented as a linked list also inherits all these properties. The creation of a stack (as a linked list) is the same as the creation of a linked list I.E, after getting a node for the ITEM to be inserted, TOP(pointer pointing to the top) points to the newly inserted node.

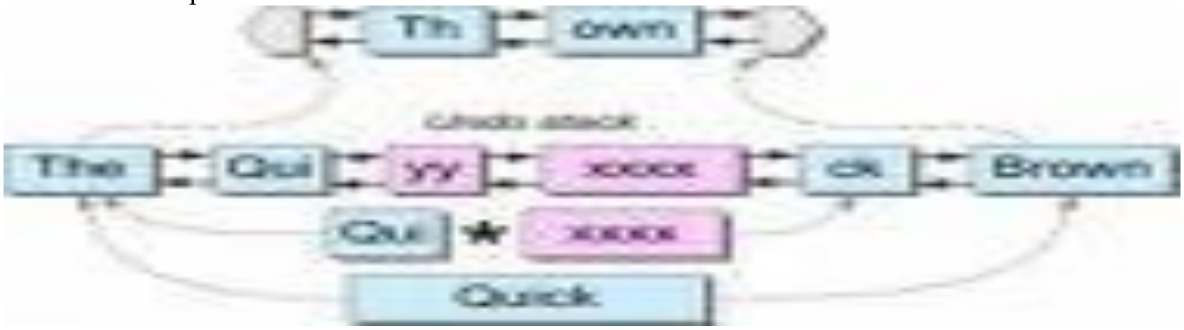
Insertion In Alinked Stack (Pushing):--

As a push can occur only at the top , top gets modified every time.
For instance, if we have a stack as shown:---



Deletion from a linked stack (popping) :--

Deletion i.e., popping also requires modification of top , that is , top is made to point to the next node in the sequence as shown:---



Solved Questions

Q1. What is the difference between linear and non- linear data structures?

Ans. Single level data structures where elements form a sequence are called linear data structures eg. Stacks, queues, linked list etc. are linear data structures.

Multilevel data structures are called non-liner data structures eg. Trees and graphs are non-linear data structures.

Q2. Describe the similarities and differences between queues and stacks.

Ans. Similarities:

(a) Both queues and stacks are special cases of linear lists.

(b) Both can be implemented as arrays or linked lists.

Dissimilarities :

(a) A Stack is a LIFO list, a queue is a FIFO list.

(b) There are no variations of a stack, a queue, however, may be circular or dequeue.

Q3. What is meant by the term "Overflow" & "Underflow"?

Ans. "Overflow" means attempt to INSERT when the list is full and no free space is available. "Underflow" means attempt to DELETE an item from an empty list.

Q4 . Distinguish between infix, prefix and postfix algebric expression giving examples of each.

Ans. Infix Notation : In this notaion, the operator symbol is placed in between the operands eg.

A+B, (A-C)* B

Prefix Notation : In this notation, the operator symbol is placed before its operands eg.

+AB, *-ACB

Postfix Notation : In this notation, the operator symbol is placed after its operands eg.

AB+, AC-B*, ABC*+

Q5 . Evaluate the following postfix notation of expression, show status of stack for each operation

500, 20, 30, + , 10, * , +

Ans. Adding] to mark the end of given postfix expression ie. 500, 20, 30 +, 10, *, +]

First 500 will be added to the stack, then 20 and then 30.

Then we get an operator + so 20 will be added to 30 and pushed to the stack. Then 10 is pushed to the stack.

Then * operator is found thus it multiplies 10 with 50 and the result is pushed into the stack.

Then again + operator is found which adds 500 with 500 that is placed in the stack. And we get the result 1000.

Unsolved Questions

Q. 1. How is computer memory allotted for a 2D array?

Q. 2. Binary search is to be used on the following sorted array to search for 30 and 60.

Index:	1	2	3	4	5	6	7	8	9	10
Value:	11	22	30	33	40	44	55	60	66	70

Give the index of the element that would be compared with at every step. Repeat the process replacing 30 by 60.

Q. 3. Consider the single dimensional array AAA [45] having base address 300 and 4 bytes is the size of each element of the array. Find the address of AAA [10], AAA [25] and AAA [40].

Q. 4. Given two dimensional array A[10][20], base address of A being 100 and width of each element is 4 bytes, find the location of A[8][15] when the array is stored as *a) column wise b) Row wise.*

Q. 5. Write a C++ function to find and display the sum of each row and each column of a 2 dimensional array of type float. Use the array and its size as parameters with float as the return type.

Q. 6. Differentiate between a FIFO list and LIFO list.

Q. 7. Transform the following expression to prefix and postfix form:

(A+B*C-D)/E*F

High Order Thinking Skills (HOTS)

Q1. What are preconditions for Binary search to be performed on a single dimensional array?

Ans. For binary search

(a) the list must be sorted

(b) lower bound and upper bound of the list must be known

Q2. How is computer memory allotted for two dimensional array?

Ans. For two-dimensional array, computer memory is allocated either in Row-major form or column-major form.

Row-major form stores the 2-D array row wise ie. firstly the first row is stored, then the second row, then the third row and so forth.

Column-major form stores the 2-D array columnwise ie. firstly the first column, then the second column, then the third column and so on.

The default form is Row-major form.

Q3. Calculate the address of X[4,3] in a two dimensional arrayX[1….5,1….4] stored in a row major order in the main memory. Assume the base address to be 1000 and that each element requires 4 words of storage.

Ans.

Base address	B=1000
Word size	W=4 bytes
No. of columns	n=4
First row number	L1=1
First column number	L2=1

Using the formula

Address X[I,T] = B + W[n(I-L1)+(J+L2)]

Given I=4, J=3

Address of X[4,3] =1000+4[4(4-1)+(3-1)]

=1000+4(4(3)+2)

=1000+4(14)

=1000+56 =1056

Q. 4. Transform the following expressions to infix form:

1. + - ABC
2. + A – BC

Q. 5. Evaluate the following postfix expression using a stack and show the contents of the stack after execution of each operation

5, 6, 9, +, 80, 5, *, -, /

Q 6. Give the necessary declaration of a linked implemented stack containing integer type numbers; also write a user defined function in C++ to pop a number from this stack.