Data Structure & Algorithm

Part -2

ATUL KUMAR (LINKEDIN)
INSTAGRAM -@codeatul

? What is Sorting? Discuss it's types.

Sorting is a method to arrange N elements of array in a particular format either ascending or descending Order.

Types :-

- Bubble Sort
- Selection Sort
- Insertion Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Redix Sort
- Shell Sort
- tree Sort
- Bucket Sort

Example :-

int a[5] = {10, 12, 8, 100, 7}

ascending → 7, 8, 10, 12, 100

descending → 100, 12, 10, 8, 7

? What is Bubble Sort ? Write algorithm & Procedure.

✓ **Bubble Sort :-** Bubble sort arrange N number of array elements by placing the biggest element on proper position. It always arrange the data in descending order.

ex → int a[5] = { 7, 6, 3, 4, 1 }
= 1, 3, 4, 6, 7.

Procedure :-

(I) 7 6 3 4 1
6 7 3 4 1
6 3 7 4 1
6 3 4 7 1
6 3 4 1 [7] → 7

(II) 6 3 4 1
3 6 4 1
3 4 6 1
3 4 1 [6] → 6

(III) 3 4 1
× 4 3
3 4 2
3 1 [4] → 4

(IV) 3 1
1 [3] → 3

(V) 1

[1 3 4 6 7]

6 3 4 1 7

6 3 4 7 1
6 3 7 4 1
6 7 3 4 1

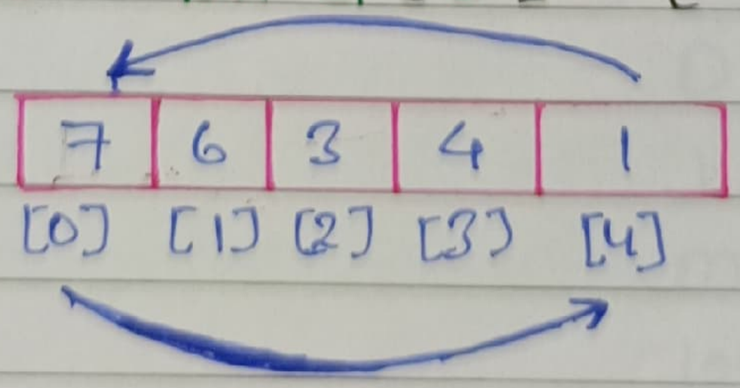
Algorithm for Bubble Sort :-

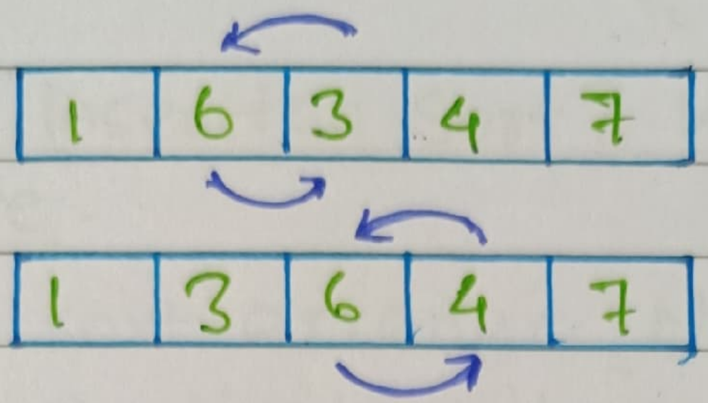
Step 1: Begin
Step 2: input a[5] 0-4 {7, 6, 3, 4, 1}
Step 3: Set i ← 3
Step 4:- Repeat Step (4) & (9) while (i >= 0) 3 >=0
Step 5: Set j ← 0
Step 6: Repeat Step (7) & (8) while (j <= i) 0 <=3
Step 7: if a[j] > a[j+1] then
Set temp ← a[j]
a[j] ← a[j+1]
a[j+1] ← temp
Step 8: j ← j+1
Step 9: i ← i - 1
Step 10: print a[5]
Step 11: exit.

? What is Selection Sort? Algorithm & procedure.

✓ Selection Sort arrange N elements of array by placing the smallest element in proper position in case of ascending order arrangement.

Procedure :- int a[5] = {7, 6, 3, 4, 1}





ascending Order.

Algorithm :-

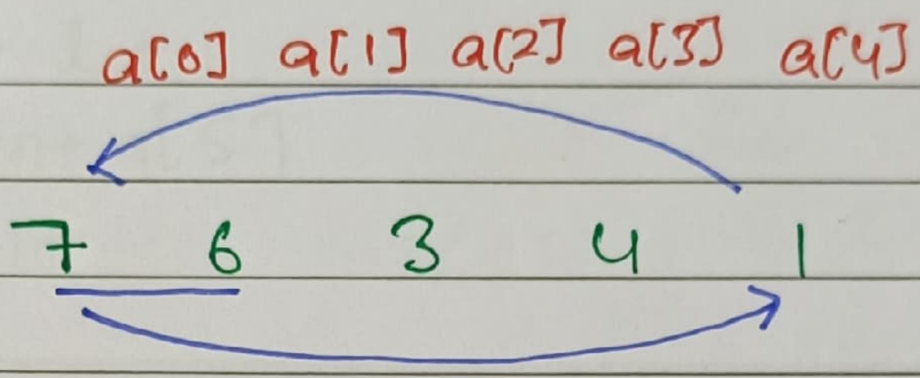
Step 1 : Begin
Step 2 : input a [5] = { 7, 6, 3, 4, 1 }
Step 3 : set i ← 0 1
Step 4 : Repeat step (5) to (10) While (i < 4)
Step 5 : set m ← a[i], i ← i + 1
Step 6 : Repeat step (7) to (8) While (j < 5)
Step 7 : if m > a [j] then
set m ← a [j]
set loc ← j
Step 8 : j ← j+1
Step 9 : if a [loc] < a [i] then
set temp ← a [loc]
a [loc]← a [i]
a [i] ← temp
Step 10 : Set i ← i+1
Step 11 : print a [5]
Step 12 : Exit.

0 = 0 1
J = 1 2 3
m = 7 6 3 4 1
loc = 1 2 3 4

? What is Insertion Sort? Write algorithm and Procedure.

Insertion sort arrange N elements of array by inserting particular item in a particular place such a way that the item are in sorted order.

Procedure :-



Algorithm.

Step 1: Begin
Step 2: input a[5] = { 7, 6, 3, 4, 1 }
Step 3: Set i ← 1
Step 4: Repeat step (5) to (9) while (i < 5)
Step 5: Set i ← 0

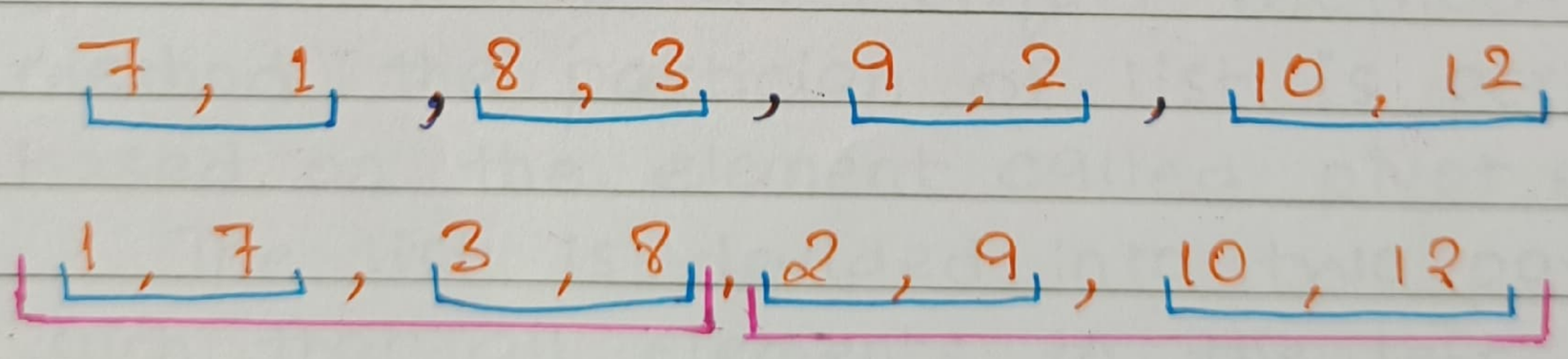
i < 5

Step 6: Repeat step (7) to (8) while $j >= 1$ (1 > 1)
Step 7: if a[j-1] > a[j] then
set temp ← a[j-1]
a[j-1] ← a[j]
a[j] ← temp.
Step 8: j ← j-1
Step 9: i ← i+1
Step 10: Print a[5]
Step 11: Exit.

? What is merge sort? Write algorithm & Procedure.

✓ Merge sort is a sort algorithm that splits the items of array to be sorted order into two group, recursively sort each group and merges then into a final sorted sequence.

Procedure :-



Algorithm :-

Step 1: Begin
Step 2: Divides all the items of array into two items pairs. IF single item remains make a single group.
Step 3: Sort each of the pair.
Step 4: Merge two pairs into a single pair.
Step 5: Sort the merge pair.
Step 6: Repeat the above steps until the all elements of array are not sorted.
Step 7: exit.

? What is Quick Sort? Write algorithm & Procedure.

✓ The Quick Sort algorithm items to Seperates the list of elements into two parts and then sort each part recursively.
It use devide an conquer method. In this method the partician of list is performed based on the element called pivot element.
The list is devided into two partician such that all elements to the left pivot are smaller than the pivot and all elements of right of pivot are greater than or equal to pivot.

Procedure :-

Pivot element.

(50)	3	1	60	65	45	90	13	67
13 →	3	1	60	65	45	90	(50)	67
13	3	1	(50)	65	45	90	60	67
13	3	1	45	65	(50)	90	60	67
13	3	1	45	(50)	65	90	60	67
Pivot (13)	3	1	45	~~50~~	(65)	90	60	67
1	3	(13)	45		60	90	(65)	67
(1)	3				60	(65)	90	67
	3						(90)	67
							67	90

1	3	13	45	50	60	65	67	90

Algorithm :-

Step 1 : Begin.
Step 2 : Select the start element of array as a pivot element.
Step 3 : Scan and find smallest element from right side of array.
Step 4 : Interchange both elements.
Step 5 : Scan and find the biggest element from left side of array.
Step 6 : Repeat the above process until all the elements of left side are smaller and right side elements are greater than pivot ~~table~~ element.

Step 7: Now, we have two sublist are available, apply same process on each sublist until the all elements of array are not started.

Step 8: exit.

Linked List

? What is Linked list? Discuss with its types.

✓ Linked List is the linear data structure where data are not stored sequentially inside the computer memory but they are link with each other by the help of address.

Example :- Data – MILAN

add		Data	Link	
1	→	(I)	(4)	
2				
3	→	N	N-1	End.
4	→	L	8	
5				
6				
7	start →	M	(1)	MILAN
8		(A)	(3)	

Types :-

- Singly Linked list
- doubly Linked list
- Circular linked list

? What is Singly linked list? Explain with algorithm.

✓ It is linear collection of data item called node, where each node has devided into two part i.e. "data part & link part", data part store the data item and link part store the address of next node.
Linked list start with a special pointer called start pointer and terminated with Null pointer.

add	Data	link
1	I	4
2		
3	N	Null
4	L	8
5		
6		
Start → 7	M	1
8	A	3

Data → MILAN

MILAN

Algorithm:- Step 1: Begin i = 7 Null
Step 2: Set i ← st 3 ≠ 4 ≠ 7 ≠ Null
Step 3: Repeat Step (4) & (5) while i ≠ Null
Step 4: Print Data[i]
Step 5: i ← Link[i]
Step 6: exit

? W.A.A. to insert an item at the beginning of Singly linked list.

Step 1: Begin
Step 2: IF fr = Null then, print Overflow and exit.
Step 3: input new item
Step 4: Set new ← fr , fr ← link [fr]
Step 5: Data [new] ← item
Step 6: link [new] ← st
Step 7: st ← new
Step 8: exit

Data - ABCD

add		Data	link
st →	1	A	2
	2	B	3
	3	C	4
	4	D	Null
fr →	5		6
	6		7
	7		8
	8		Null

Data → ABCD

add		Data	link	
st →	1	A	2	
	2	B	3	
	3	C	4	
	4	D	Null	
new ~~fr~~ →	5	X	~~6~~ 1	← st
~~fr~~ →	6		7	
	7		8	
	8		Null	

? W.A.A. to insert an item at the ending of Singly linked list.

Step 1: Begin.
Step 2: if fr = Null then
Print Overflow and exit.
Step 3: input new item.

Step 4 : Set new ← fr , fr ← link [fr]
Step 5 : set loc ← st
Step 6 : Repeat Step (4) while link [loc] ≠ Null
(4) loc ← link [loc]
Step 7 : data [new] ← item
Step 8 : link [new] ← Null
Step 9 : link [loc] ← new
Step 10 : exit

Data → ABCD

add	Data	link
st → 1	A	2
2	B	3
3	C	6
4	D	Null
fr → 5		6
6		7
7		8
8		Null

loc → 1 2 3 4

Data - ABCD

add	Data	link
st → 1	A	2
2	B	3
3	C	4
4	D	5 Null
new fr → 5		Null
fr → 6		7
7		8
8		Null

? What is Doubly linked list ? With algorithm...

It is the linear collection of data item called node where each node has derived into three part i.e. data part, previous part and next part.

Data part store the data item, previous part store address of previous node and next part store address of next node. It start with special pointer called first pointer and ending with last pointer.

	add	Previous	Data	next
first →	1	Null	A	2
	2	1	N	3
	3	2	I	4
	4	3	S	5
Last →	5	4	H	Null
	6			7
	7			8
	8			Null

① Algorithm for forward traversing:

Step 1: Begin
Step 2: Set i ← first
Step 3: Repeat Step ④ & ⑤ while i ≠ Null
Step 4: print data [i]
Step 5: i ← next [i]
Step 6: exit

ANISH
1 2 3 4 5 Null

	add	Previous	data	next
first →	1	Null	A	2
	2	1	N	3
	3	2	I	4
	4	3	S	5
Last →	5	4	H	Null
	6			
	7			
	8			

(2) Algorithm for backward traversing.

Step 1: Begin
Step 2: Set i ← last 5
Step 3: Repeat step (4) & (5) while i ≠ Null
Step 4: print data [i]
Step 5: i ← previous [i]
Step 6: Exit

? W.A.A. to insert an element at the beginning of doubly linked list?

Algorithm :-

Step 1: Begin
Step 2: if fr = Null then
print Overflow & exit.
Step 3: input new item.
Step 4: set new ← fr , fr ← next [fr]
Step 5: data [new] ← item.
Step 6: next [new] ← first.
Step 7: previous [new] ← last
Step 8: first ← new
Step 9: exit

(I)

		Previous	data	next
first →	1	Null	A	2
	2	1	B	3
	3	2	C	4
Last →	4	3	D	Null
Fr →	5			6
	6			7
	7			8
	8			Null

(II)

		Previous	data	next
First →	1	Null	A	2
	2	1	B	3
	3	2	C	4
Last →	4	3	D	Null
new ~~Fr~~ →	5	4	X	~~6~~ 1
Fr →	6			7
	7			8
	8			Null

f → ABCD
B → DCBA

f → XABCD
B → XDCBA

? What is Circular linked list? And its types.

✓ The Circular linked list is the variation of linked list where the first node point to last node and last node point to first node.

add	Data	link
st → (1)	A	2
(2)	B	(3)
(3)	C	4
(4)	D	(1)
5		
6		

(I) ABCD -----
(II) (III) (IV) -----

add	Data	link
st → (1)	A	(2)
2	B	3
3	C	4
4	D	Null
5		
6		

Null → terminate.

ABCD

types :-

- Singly Circular linked list
- Doubly Circular linked list.

? Circular Singly linked list :-

✓ If a last node of Singly linked list hold the address of start node then the linked list is called Circular Singly linked list.

Ex -

add	Data	link
Start → 1	A	2
2	B	3
3	C	4
Last → 4	D	1

add	Data	link
1	A	2
2	B	3
3	C	4
4	D	Null → end
5		
6		

ABCDABCD.....

ABCD

Algorithm :- Step 1 : Begin

Step 2 : Set i ← st

Step 3 : Print Data [i]

Step 4 : i ← link [i]

Step 5 : go to step (3) while i ≠ st

Step 6 : exit.

1 2 3 4 1

ABCD

3 2 ≠ 1

1 ≠ 1 X

? W.A.A. to Print Content of C.S.L. more than One time.

Algorithm -

Step 1 : Begin

Step 2 : Set i ← st , j ← 1

Step 3 : Repeat step (4) to (7) while j <= 2

X 2

Step 4: Print data[i]
Step 5: i ← link[i]
Step 6: if i = st then
Step 7: j ← j+1
Step 8: exit

add	Data	link
st → 1	A	2
2	B	3
3	C	4
4	D	1
5		

ABCDABCD

? What is Circular doubly linked list?

? W.A.A. for forward and backword traversing of circular doubly linked list.

✓ Circular doubly linked list :-

If a last node of doubly linked list held the address of first node and first node of doubly linked list hold the address of last node then, the linked list is called Circular doubly linked list.

add	Previous	data	next	
First → 1	~~X~~ Null	A	2	
2	1	B	3	
3	2	C	4	ABCD
Last → 4	3	D	~~X~~ Null	DCBA
5				End.

? W.A.A. for forword traversing of Circular doubly linked list.

✓ Algorithm :-

add		Previous	data	next
First →	1	4	A	2
	2	1	B	3
	3	2	C	4
last →	4	3	D	1
	5			
	6			

Step 1: Begin ← i = X 3 4 1

Step 2: Set i ← first

Step 3: print data [i] ABC 1 ≠ 1

Step 4: i ← next [i] A Z Z ≠ 1

Step 5: go to step (3) while i ≠ first

Step 6: exit

? W.A.A. for backward traversing of Circular doubly linked list.

✓ Algorithm

add		Previous	data	next
First →	1	4	A	2
	2	1	B	3
	3	2	C	4
last →	4	3	D	1
	5			
	6			

Step 1: Begin
Step 2: set i ← last 4
Step 3: print data [i] DCBA
Step 4: i ← previous [i] ~~3 ≠ 4~~ 3 ≠ 4
Step 5: go to step (3) while i ≠ last
Step 6: Exit

? Difference between Array and linked list.

Array	Linked list
(I) Array is a collection of homogenous (Similar) data type.	(I) Linked list is a collection of node (data & address).
(II) Array elements are stored in continous memory location.	(II) Linked list elements can be stored anywhere in memory.
(III) Array work with static data structure.	(III) Linked list work with dynamic data structure.
(IV) Array elements are independent to each other.	(IV) Linked list elements are dependent to each other.
(V) Array takes more time. Ex - insertion, deletion etc.	(V) Linked list takes less time. Ex - Insertion, deletion etc.