

-	
	Unit 5
*	Julie :- Linear data structure in which invertion is done at the end of point pla the rear and
	is done at the end of point & a the rear and
7	deletion at the first and point. 12 de front.
. 1 80 4.1.3	Front and rear are frem pointers.
	1 sort mac star art. garas present
10- 1	2 operations of queue in arxay:
1.	2 operations of queue in array:
ચ.	Delete
	- Constant to Manage to
	GINSERT (Dune, N, Front, Rear, Ele)
	If Rear = N then I have a like . I have a like
	write: overflow and evit.
	If Front = 0 there
	Front := Rear: = 12 00 00 1 = 2 3 50 2 12
	else él manal de la compa de la constitue
	rear: = rear +)
	B. Set Fart: = WATHET.
3.	Set June [rear]: = Ele
	M. Restriction.
٩.	Enit-
2	P DELETE (June, N, Front, Rear, ele)
1.	If Front = 0 then
	write: underflow and enit
۵.	Set element! = Queul [Front]
N. A.	Sannad by CamSannar

3.	If front = Rear
	If front = Rear Set front := rear := 0.
	The state of the s
	else
	Set front := front +1
Ч.	Exit.
	The speciment of the state of t
*	Circular Julue: - In circular que all nodes are
	treated as circular. Last node is connected back to the
	first node. Cocular queue is also k/a Ring Buffer.
•	It is an abstract data types.
•	Circular quine contains a collection of data which allow
	insertion of data at the end of the queue and deletion
	of data at the beginning of the queue.
	Harrie U
	2 operation of cércular queue in an array:-
1.	Thert Flimme School Charles &
ک.	Delete Artis Sons Stelen . worth maken warmer with
	can be much at best some seat is second inset
	Insert (Queue, N, Front, Rear, ele)
	If (FRONT=1 and Rear = N) or (Front = Rear +1)
	wrête: Overflow and enit.
2.	Ale throat - O from
	If front = 0 then front: = Rear := 1
	else
	rear: = rear +1
3.	Set Queue [Rear]: = Ele.
Mark.	Enit

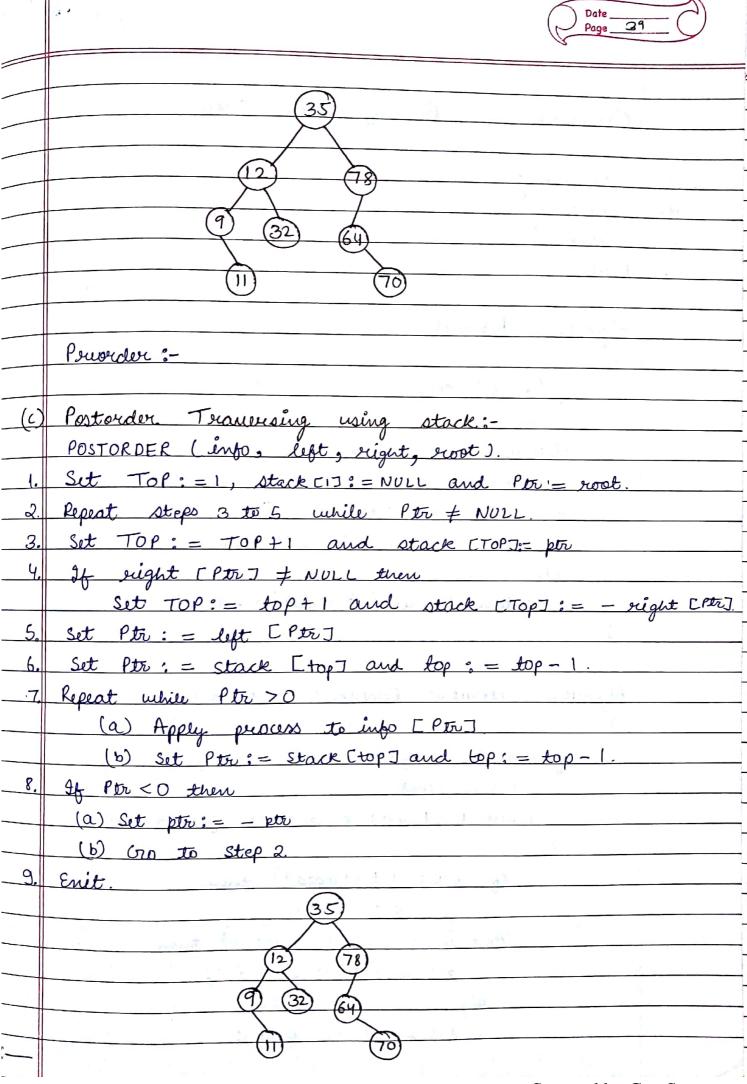
2.	DELETE (Queue, N, Escont, Reason, Ele)					
1.	If front = 0 then					
E	write: Orderflow and enit					
	and a contract of the contract					
ಎ .	Set ele: = Pueue [Front]					
	jesses Cirarias					
3.	If front = Rear					
-	Set front: = Rear: =0					
0,500	of the most street in the contraction of the contraction of the					
1.0	else if front = No then I deliver an deliver					
	Set front := 1 miles make the state of the					
	the and the date of the					
u.j.	- else hope in minerial a whaters are in mineral in					
la i ti t	front := front +1					
	and the first that the state of					
ч.	Enit.					
	integral to the formation of the decay of					
*	DE Julie: - (Double ended. Julie):>					
	In double ended Jume, insert and delete operation					
	can be occur at both ends that is front and					
	near of the queue.					
<u>() </u>	. It is (FROM I and Deep of Deep I Front o Fee.					
	There are 2 types of DE June:					
1.	Input restricted Queue:-					
•	It is a dequeue, which allows insertion at only I					
	end, rear end,					
•	It allows deletion at both ends, rear and front					
	end of the list.					
	1					
	Delote Insert					
	front Rear					

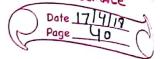
2.	Output restricted queul:					
•	It is a dequeur, which allows deletion at only one end					
	front end					
git of the	It allows invertion at both ends, near and quant ends, of					
*	the list Insert					
	Twent					
	Delete Front Rear					
*	Priority Jueue:-					
1,000	for Priority que items are ordered by key values					
	so that item with the lowest value of key is at					
Ú	front and item with the nighest value of key					
	is at year or vice wasa.					
ciority	Quine 2 -3 4 5 6 7 8					
0	The a first AA dame to have a wide					
2	BB CC DOWN LL					
3	MM					
4	: Some (RHAY) and KKE and (Reports " secur war of)					
5	when the west word (some) there EE GG PP					
6	II 11					
	alt water and the sail that all all the					
X	tagen letter wir wire wellt					
^						
	(11) The equipment of the second of the contract of the contra					
	The second of th					
	grante rate une metidad catalista des especialista (VI)					
7.0	The state of the s					

*	TREE: - It is a non-linear data structure.
	Tree represents the nade connected by edges.
ر ا	Binary tree: - Binary Tree is a special datastructure used for data storage purpose. A binary tree is
	used for data storage purpose. A benary true is
	a special condition that each node can have a
	manipular of two children.
	* Lever Target grant-
૨.	complete binary tree: - A complete binary tree is
	a tree that is completely filled with the possible
- L	enception of the bottom level. The bottom level is
	filled from left to right.
10	Binary Search True? - A binary search tree is
	a binary tree it may be empty. If it is not
	a binary tree it may be empty. If it is not empty then it satisfy the following properties.
(i)	envey node (element) has the key (value) and no
1	two element (node) have the same value.
	्राप्ता करणा करणा करणा करणा करणा करणा करणा करण
(ii)	The value in the left sub tree are smaller than
	the value in the root.
(iii)	The values in sight sub tree are greater man
	the value in the root.
(iv)	The left and right sub true are also kinary
رس	Slarch tru.
	See our party

	Representation of a binary true:-
1.	Arvay
۵.	linked list.
-X-	
	Traversing in Binary Search tree:-
,	Grorder Trauvising Left Node Right
ઢ.	Pre order Traversing Node Left Right
3.	Pre order Traversing Left Right Nade
	Inorder Traversing using recursion:
	Void inveder (node * Jost)
	{ if (root ! = NULL)
	{
	inorder (root -> left);
	printf ('10d', root -> info);
	inorder (root → right);
	3
	to the second se
	the state of the state of the state of the state of
	Traversing in a Binsey tree:
	and the state of t
(a)	Gnorder Traversing using stack:-
	INORDER (Info, left, light, root).
	Set top:=1, Stack [1]:= NULL and Ptr:= start.
ર.	Repeat while pto 1 = NULL
	a) Set Top: = Top+1 and stack [top]:=ptr.
	b) Set Ptr: = left [P.tr.].
3.	Set Ptr: = Stack [Top] and Top: = Top-1.
4.	Repeat steps 5 to 7 while Ptr \$ NULL
5.	Apply process to into EPtri]
6.	If right [Ptr] \$ NULL then
-	The state of the s

1	
	(a) Set Ptr: = right CPtr]
	(b) (no to step 2
7.	Set Ptu: = stack [Top] and Top: = Top-1.
8.	Enit.
	= (35)
	(12) (78)
	(g) (64)
	(70)
	Inorder: - 9, 11, 12, 32, 35, 64, 70, 78.
()	Parandar Taranasa atanbar
_(b)	Preorder Traversing using stack:- PREORDER (Into, left, right, root)
-	Set top:=1, Stack [1]:= NULL and Ptr:= stack
ચ .	Repeat step 3 to 5 while Ptr = NULL
3.	Apply prous to Info [Ptr]
ч.	If right [Ptr] & NVLL then
	Set top: = top+1 and Stack [top]:= right CPtr.]
5.	If left [Ptr] = NULL then
J	Set Ptr := left CPtr]
	else
	Set ptr:= Stack [top] and top:= top-1.
6.	Enit.
1 1 -	





	Operations on Binary search tree:-
	Transpara
,	Traversing Searching
	Insertion
	Deletion
	(1) (1)
	Algorithm Rsearch
	A 100 100 100 100 100 100 100 100 100 10
	if (t=0) then
	return 0;
	else if (n=t → data) then
	return t:
	else if (n < t > data) then
	return Rsearch (t) lichtld, x);
	else se la companya de la companya d
	return Rearch (t -> recheld, x);
	3
	Algorithm iterative besich (root, x)
	1 The state of the
	found: = false
	t:= root
\parallel	while ((t \$0) and not found) do
\parallel	A Appendix of the Control of the Con
	if (n = (t → data)) then
	found: = true;
1	else if (n < (t -> data)) then
\parallel	$t:=(t\rightarrow larild);$
~ i	else (a) (b)
	t: = (t - rehild);
	}

if (not fo	und) then
retur	
else	
return	i t:
3	
Algorithm	Insertion (root, 21)
{	
found :=	= false;
V	root;
while	(P = Null and not found) do
{	v
	$q_{i} = \rho$
	if (n = (P → data)) then
	found: = true;
	else if (n ∠(p → data)) then
	$p:=(p \rightarrow Achild);$
	else
	$p:=(p \rightarrow xehild);$
3	
i	f (not found) then
· · · · · · · · · · · · · · · · · · ·	p:= new Treenode;
4 12 101	(P→ lchild): = NULL;
	(p → sichild): = NULL;
	$(P \rightarrow data) := \mathcal{X};$
	if (root \neq NULL) then
	in factor to the latest the lates
	if (n < (q → data)) then (q → louid):= p;

		Date Page
else	ac to 1 Mar was 150	
	$q \rightarrow rchild):=p;$	
3	T - 700 min j	
else	A contract	
	loot:=p;	
3		
3	CA SERVE WATER	1-1-1
		, ? _ ?
		المرة
	with the boats death to a first	
- 10 14	Carl agrant agrant	
7.		
	5 10	
	C-11 25 C C1	
	Anna de la companya d	