· Arrays :-

· Majorily Element (Moore's voting algorithm):

Time complainty
$$\rightarrow O(N) \rightarrow \text{space}(0)(1)$$
 1 4 3 2 4 4 4 2

pseudo coole \longrightarrow element = away [0]

count = 1;

Here we check the next element if (element != away [i])

if it is not then we use count = :

declement declement can be solved using element = away [i]

7 This Problem can be solved using element = away [i]

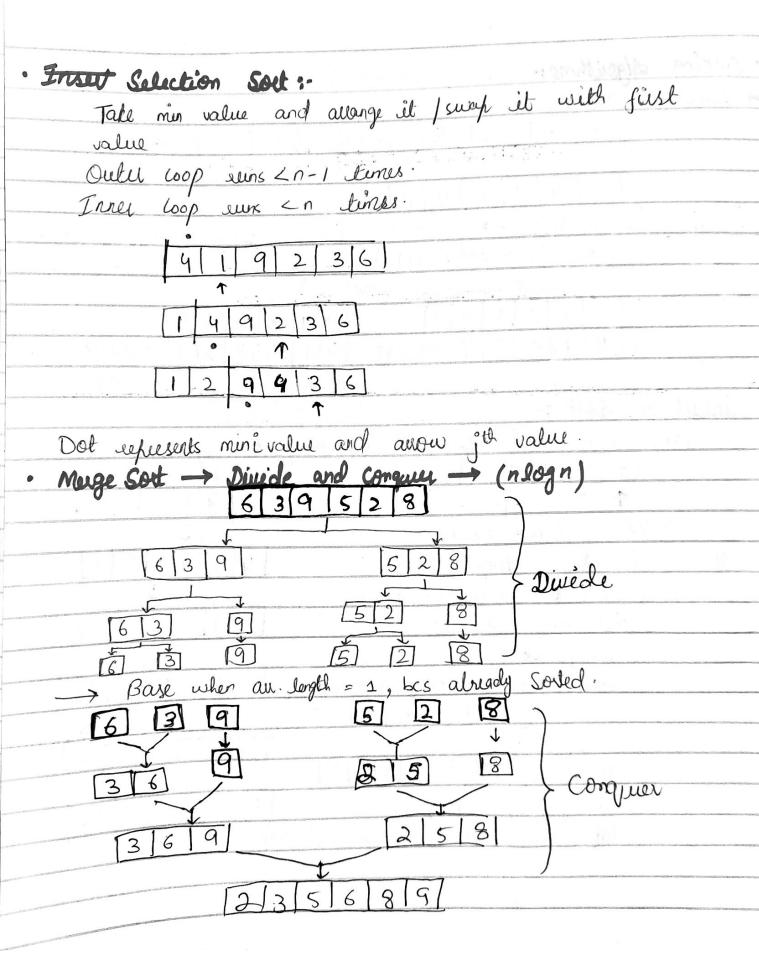
snap it time complainty is count + t

O(N) \rightarrow space $O(N)$.

Largest Sun Contigues Subarray (Kadare's algorithm): " if a negative number occurs we move to the next element and stoll sum. function carged Sum (array) macSun = 0, cussur = 0; for (let i= 0; i away length; i++) cullim = culsium + array [i] if (cursum > marsum) may sum = cuesum; if (cursum × 0) -> 12 -> madsum cu = 0; return mar Sun; · Check sun Zero (duay must be sorted):-Approach: Left = 0; Right = away length - 1, sun; sum = a [left] + a [right] if sun > 0 right -else left ++; Sum == 0; console-log.

· Best time to Buy & sell s	tocks -> LostCod	2-121.
3 3 1 7 4		in antonion of the
you can buy only one ti	ne:	A. Observation
min = Math. min (min, ali	J);	ang William
max = Math. max (max, ali		
after loop the return		
eltuen man-min;		
The second secon		NAME OF THE PROPERTY OF THE PARTY OF THE PAR
· Best time to Buy & sell.	stocks - Leel	Code 122:-
	1000 000 0	
5 2 7 3	6 1 2 4	11
Let profit = 0;		111
if (areay [i] > array [i-1])	77	7
plofit += anay [i] - a	vay (i-1);	5 1/35
return profit;	2.00 2.00	5/ 1/
		3 3
Trapping Roin water- lead	Code 42:-	2
13 11 2 1 4 1 0 1	113 2	5+3+3=11
Logic: water well only be teappe	d when o	
1 It William Conditions	01 2 11 3	20
will be greater.	1 8	ic o
Left -	5 10 12	2 - Right
	131 + 5	
	Left-Mar = 3, 3, 3	4, 4, 4, 4, 4
	aght Mad = 4, 4, 4	, 4, 3, 3, 3, 2
The second secon	value = Min(Left	[i], sight (i))-a [i]
THE RESIDENCE AND ADDRESS OF THE PARTY OF TH		

Suppling is used and the largest element will move to the end. S 4 1 5 9 2	· sorting Algorithms:-	- to the section to the section of
end. $8 \ $	- Bubble sort:	
end. $8 \ $	Swapping is used and the largest	clement will move to the
in first idealism \rightarrow i = n \rightarrow j = 0; j <= i; j+t if (awy [j] > away [j+1]) Swaf (away [j], owly [j+1]) [4 8 1 5 9 2 4 1 8 5 9 2 4 1 5 8 9 2 4 1 5 8 2 9 \rightarrow Larget at end. insultion Solt: Take away and shlit it in two parts tomp 8 4 1 5 9 2 Solted non-sorted. Store clament in temp = 48 1 5 9 2 for (let i=t; i < n; i+t) let lemp = a [i]; while (j >= 0 & & a [j] > a temp) & 1 4 5 8 9 2 away (j+1] = away (j) [1 4 5 8 9 2	end.	
in first idealism \rightarrow i = n \rightarrow j = 0; j <= i; j+t if (awy [j] > away [j+1]) Swaf (away [j], owly [j+1]) [4 8 1 5 9 2 4 1 8 5 9 2 4 1 5 8 9 2 4 1 5 8 2 9 \rightarrow Larget at end. insultion Solt: Take away and shlit it in two parts tomp 8 4 1 5 9 2 Solted non-sorted. Store clament in temp = 48 1 5 9 2 for (let i=t; i < n; i+t) let lemp = a [i]; while (j >= 0 & & a [j] > a temp) & 1 4 5 8 9 2 away (j+1] = away (j) [1 4 5 8 9 2	8 4 1 5 9 2	
insuktion Sort: Take away and split it in two parts temp 8 4 1 5 9 2 Sorted non-sorted. Store clanarl in temp = 48 1592 for (let $i=1$; $i=n$; $i+1$) let temp = a [i]; while $(j>=0$ 5 5 6 1 1 1 1 1 1 1 1 1 1	in first iteration -> i=n -> j=	= 0 ; j <= ů ; j++
Swood (away (j), only (jt) 4 8 1 5 9 2 4 1 5 8 9 2 4 1 5 8 2 9 \rightarrow Larget at end. Insultion Solt: Take away and split it in two parts temp 8 4 1 5 9 2 Souted non-sorted. Shore clamant in temp = 48 1592 for (let i=1; i <n; (j="" [i];="" i+t)="" let="" temp="a" while="">= 0 8 4 a [j] > a temp) £ 1 4 5 8 9 2 away (j+1] = away (j) [1-2] 1 4 5 8 9 2</n;>	if (alley [j] > alley [j+1])	
4 8 1 5 9 2 4 1 8 5 9 2 4 1 5 8 2 9 $\rightarrow \text{ Larget}$ at end insultion Solt :- Take away and split it in two parts temp 8 4 1 5 9 2	swap (away (i), asky (i	+1]
insultion Sort: Take away and shift it in two parts temp 8 4 1 5 9 2 Souted non-sorted. Store element in temp = 48 1592 for (let i=1; i < n; i+t) let semp = a[i]; while (j >= 0 8 4 a[j] > = temp) & 1 4 5 8 9 2 away (j+1] = away (j) 1 4 5 8 9 2	4 8 1 5 9 2 1	118592
insultion Sort: Take away and shift it in two parts temp 8 4 1 5 9 2 Souted non-sorted. Store element in temp = 48 1592 for (let i=1; i < n; i+t) let semp = a[i]; while (j >= 0 8 4 a[j] > = temp) & 1 4 5 8 9 2 away (j+1] = away (j) 1 4 5 8 9 2	415892	1 5 8 2 (9) -> Larget at
Take away and shift it in two parts temp 8 4 1 5 9 2 Sorted non-sorted. Store clament in temp = 48 1592 for (let i=t; i < n; i+t) let temp = a[i]; while (j >= 0 9 9 a[j] > 0 temp) [145892 away (j+i] = away (j) 145892		end.
Take away and split it in two parts temp 8 4 1 5 9 2 Sorted non-sorted. Store clament in temp = 48 1592 for (let i=t; i < n; i+t) let temp = a[i]; while (j >= 0 9 9 a[j] > = temp) [145892] away (j+i] = away (j) 145892	insukion sout:	*
Souted non-sorted. Store elament in temp =	Take alsay and split it in e	two parts tomb
Sorted non-sorted. Store element in temp = 48 1592 for (let i=t; i < n; i+t) let temp = a[i];	8 4 1 5 9 2	, ,
Store element in temp = 48 1592 for (let i=t; i < n; i+t) let temp = a[i];	\uparrow \uparrow \uparrow	8 4 1 5 9 2
Store elament in temp = $\frac{481592}{1481592}$ for (let i=1; i <n; (j="" i+t)="" let="" temp="a[i];" while="">= 0 \(\xi \xi \) alg z \(\xi \) temp) \(\xi \) 4 \(\xi \xi \) 3 \(\xi \xi \) 2 \(\xi \xi \xi \xi \xi \xi \xi \xi \xi \xi</n;>		1
for (let $i=1$; $i < n$; $i+t$) let $semple = a[i];$ $semple = a[i];$ while $(j > = 0 = 2 = a[j] > = temp)$ [145892] $semple = a[i];$		481592
let $dem f_{i} = a[i];$ let $j = ie1;$ while $(j > = 0 = 2 = a[i] > = temp)$ [145892] away $(j+i] = away (j)$ [148592] [145892]	4	1
Jet $j = i = i = i = i = i = i = i = i = i = $		148592
while $(j > = 0 = 2 = a + a = 1) = a + a = 1 = a + a = 1 = a + a = 1 = a = 1 = a = 1 = 1 = a = 1 = 1 =$		14
any (j+1) = anay (j) [145892]	,	mp) [145892
3 (1) 8 1 5 9 2		, ,
3 1 592 assay (3+1) = temp' 11245892	(1;)	1145892
aslay (i+) = temp' [1]2 4 5 8 9 2	3 1 8 1 5 9 2	
	alley (i+) = temp'	1245898
bus of while loop we incemented one time.		ed one dime.



Ouick Sort :-Pivot. Pivot and Partion random Start = 0 end = n-1 mid fiest i = Start - 1; last. Pinot. Travesing for (let j= start; j'c end; j'++) s if (away [i] < pivol) Loop will end at j=42 8 Swap; swap (assay send), assay si] return i; bes condition is false. function quick soit (only) Swaped when i reached at pivot Irdon = partition (acis, end); the index -1 times from higher position quick Sort (ac), start, pivot-1); quick Sort (ac), pluot+1, end);

)