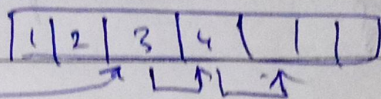


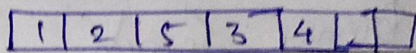
Operations On Array.

- Primary
- Traversal \rightarrow visiting every element of array
 - Insertion \rightarrow Inserting an element in array. Best case $O(1)$
Worst case $O(n)$
 - Deletion \rightarrow Deleting an element from array
 - Searching

\rightarrow Insertion.



5 at index 2



Method 2

 Index \rightarrow 2
 (Sequence doesn't matter)

Best Case :- $O(1)$

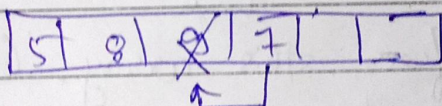
{ Insert element at end }

Worst Case :- $O(n)$

{ Insert element at index }

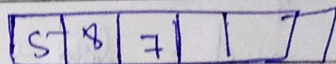
size $+ 1$

\rightarrow Deletion.



Size $- 1$

Index $- 2 \rightarrow$ delete



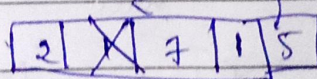
Best Case :- $O(1)$

No shifting
 { Deleting last element }

Worst Case :- $O(n)$

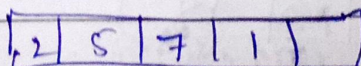
{ Deleting first element }

Method 2 :-

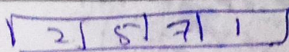


Index $\rightarrow 1$ delete

(Sequence doesn't matter)



Size $- 1$



Searching

→ Linear Search → Worst Case :- $O(n)$

Search 8 from this array

5	9	8	7	1
---	---	---	---	---

↑ ↑ ↑
8 ≠ 5 8 ≠ 9 stop
 8 = 8

{Nm sorted}

from $i=0$ to n , checking

→ Binary Search → Worst Case :- $O(\log n)$

Array is sorted.

↑ low high

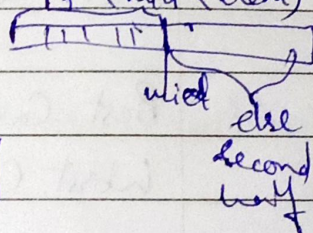
Search 8 from this array. (elem be element)

If ~~2~~ elem \leq high

$$\text{mid} = \frac{\text{low} + \text{high}}{2}$$

first half

if (ngd < elem)



7	10	18	21	33	99
↑		↑			↑
low		mid			high

$$18 > 8 \quad (7 \neq 8 \text{ or } 8 \neq 7)$$

7	10	18
---	----	----

q q q
low mid high

8701

~~Adopt~~

low ~~#~~ = high \rightarrow stop