

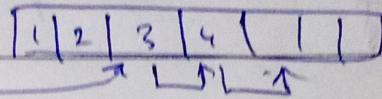
H

Operations On Array.

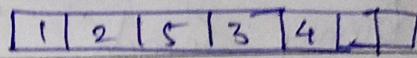
Primary

- Traversal + visiting every element of array
- Insertion + 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 Best Case :- $O(1)$

{ Insert element at end }

Worst Case :- $O(n)$

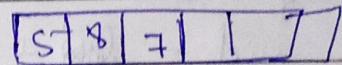
{ Insert element at index }

$$\text{size} = 1$$

\downarrow
Index 1 \rightarrow 5
 \downarrow
Index 2 \rightarrow 5
 \downarrow
Index 3 \rightarrow 5
 \downarrow
Index 4 \rightarrow 5
 \downarrow
Index 5 \rightarrow 5
 \downarrow
Index 6 \rightarrow 5

(Sequence doesn't matter)

Index -2 \rightarrow delete



$$\text{size} = 1$$

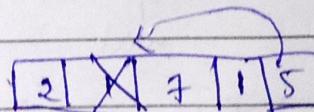
Best Case :- $O(1)$

{ No shifting
Deleting last element }

Worst Case :- $O(n)$

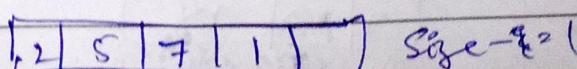
{ Deleting first element }

Method 2 :-

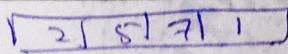


Index \rightarrow 1 delete

(Sequence doesn't matter)



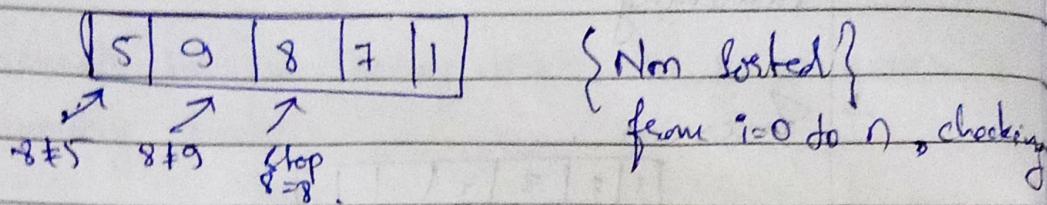
$$\text{size} = 1$$



Searching

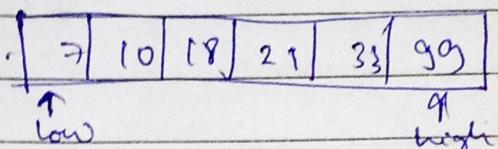
→ Linear Search → Worst Case :- $O(n)$

Search 8 from this array



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

Array is sorted.

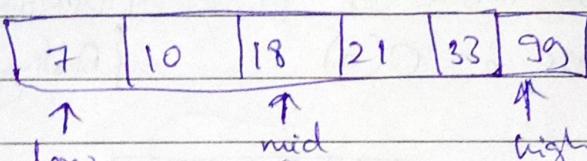


Search 8 from this array. (elem be element)

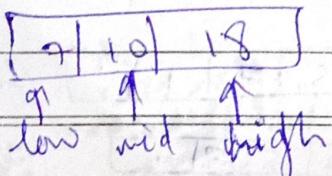
If ~~elem~~ elem < high.

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

first half
if ($\text{mid} < \text{elem}$)
mid
else
second half



$18 > 8$ (7 & 8 are 8)



~~10 > 8~~
~~(low = high → stop~~