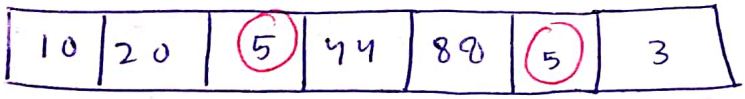
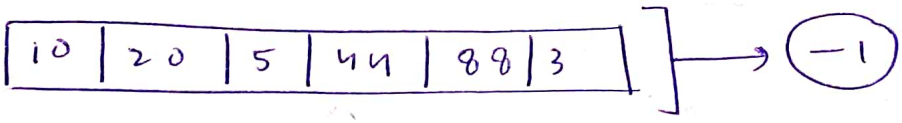
CHECK DUPLICATE





Note

- Hume iss Ques ko $O(n^2)$ me nahi krna Solve.
- Hume $O(n \log n)$ me Solve krna hai!

- $O(n \log n)$ → T.C merge Sort ki hoti hai
  - we will use merge sort.

For NOW,
we don't know Merge Sort
∴ we will use Function

FUNCTION

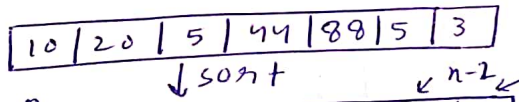
`Arrays.sort(arr)`

Yeh Function internally use hota hai!

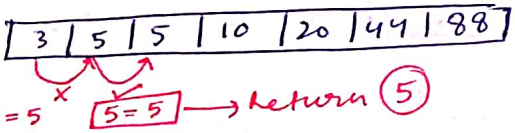
Tim Sort → Insertion Sort + Merge Sort

Sorting Algorithm based on Insertion + Merge Sort.

∴



Now we will compare each element with it's next element are they equal. if yes the return that element




```
public static void main(String [] args){
Scanner s = new Scanner(System.in);
int n = s.nextInt();
int[] arr = new int[n];
for(int i=0; i<arr.length; i++){
    arr[i] = s.nextInt();
}
Array.sort(arr);
int dup = -1;
for(int i=0; i<= arr.length-2; i++)
{ if(arr[i] == arr[i+1])
    { dup = arr[i];
      break;
    }
}
System.out.println(dup);
}
```

Time complexity

$n \log n + n = n(\log n)$

(Sorting) (Comparison)