Arrays

1st data structure

Data Structure → structures which are used to store the data.

→ Data is the fuel for the development.

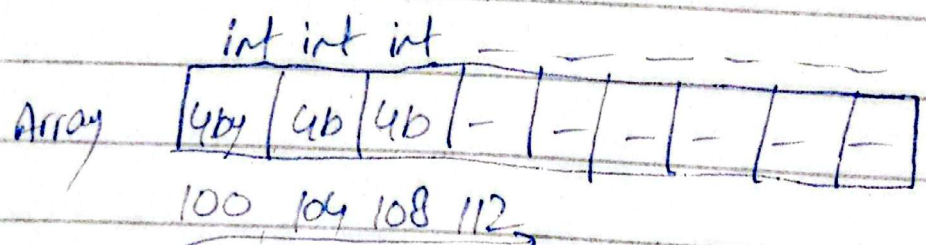
→ Different forms of data are stored in different types of structures.

Algorithms →

Methods to perform different types of operation on the data.

like searching, sorting etc.

Array Creation



① Same type of data can only be stored.

② It is contiguous in memory.

③ Linear

```
int marks[5]
 |     |     |
Data   Name  Size
type
```

Declarations.

int marks[5] = {10, 20, 30, 40, 50}
Limiting the size to 5

int marks = {10, 20, 30, 40, 50, 60, 70, 80}
No size limit as we have not specified anything

10	20	30	40	50
0	1	2	3	4

Index

Supposed example outcome:
marks[1] = 20

Loops in Array

```
for(int i=0; i<size; i++){
    cout << marks[i] << endl;
}
```

size of array can be found by:

Size of [array] / Size of [int] = Actual size of array

It will be in individual binaries.

Find the Smallest Number of Array :-

int sum = 0 → INT_MAX

Logic:

```
if(nums[i] < smallest){
    smallest = nums[i];
}
```

smallest = min(nums[i], smallest)

largest = max(nums[i], largest.

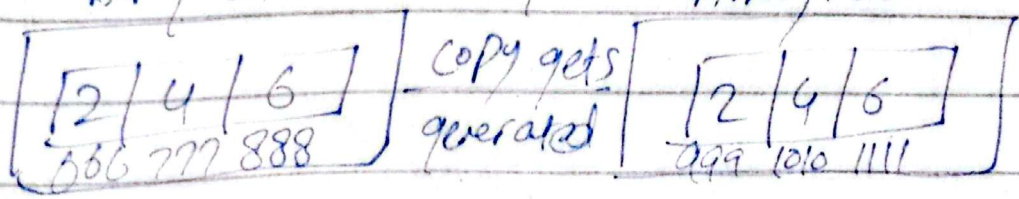
Pass By Reference

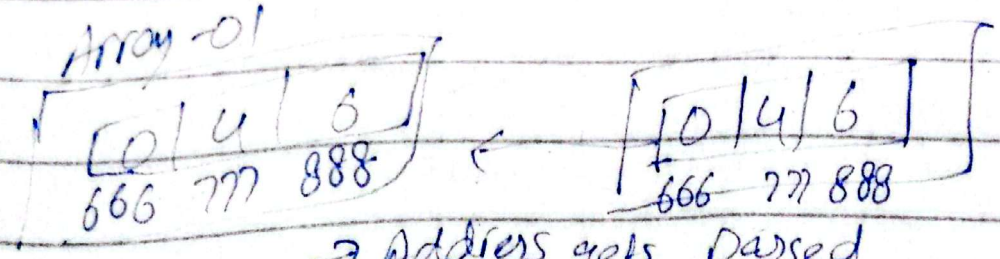
↓

address.

Address gets pass directly as

Pass By value





→ Address gets passed

→ Copy do not gets created

→ change in one reflects

→ Linear Search

Target variable

↓

To be found

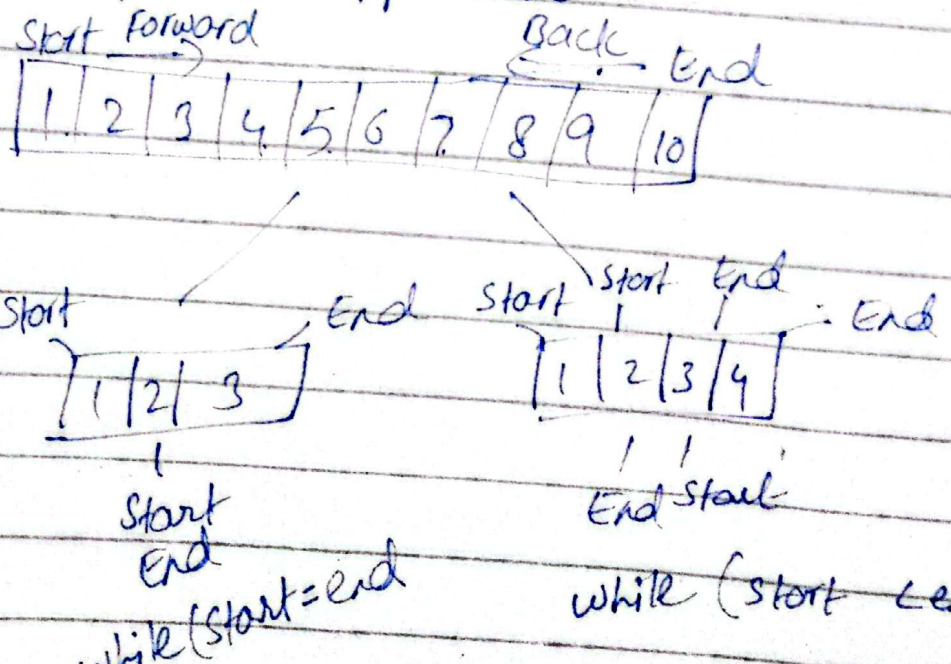
```
If (marks[i] = target){
    return i;
}
```

Time Complexity

gives us the Binary Search method as used which is used for searching purpose.

→ Reverse an Array

2 pointers approach



① Sum & Product of array

② Swap the max & min of array

③ print all unique number

④ Intersection of 2 arrays