Array

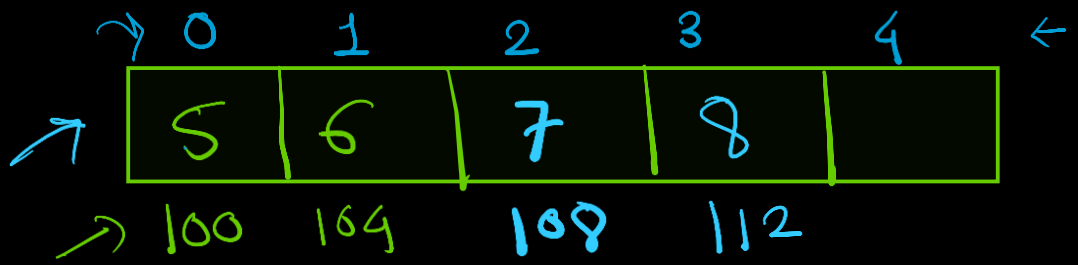
- What is Array ?
- Types of Array
  - 1 D array
  - Multi Dimensional Array
  - jagged Array
- Operations
  - Insertion
  - Deletion
- Searching
  - Linear Search
  - Binary Search (To Do).

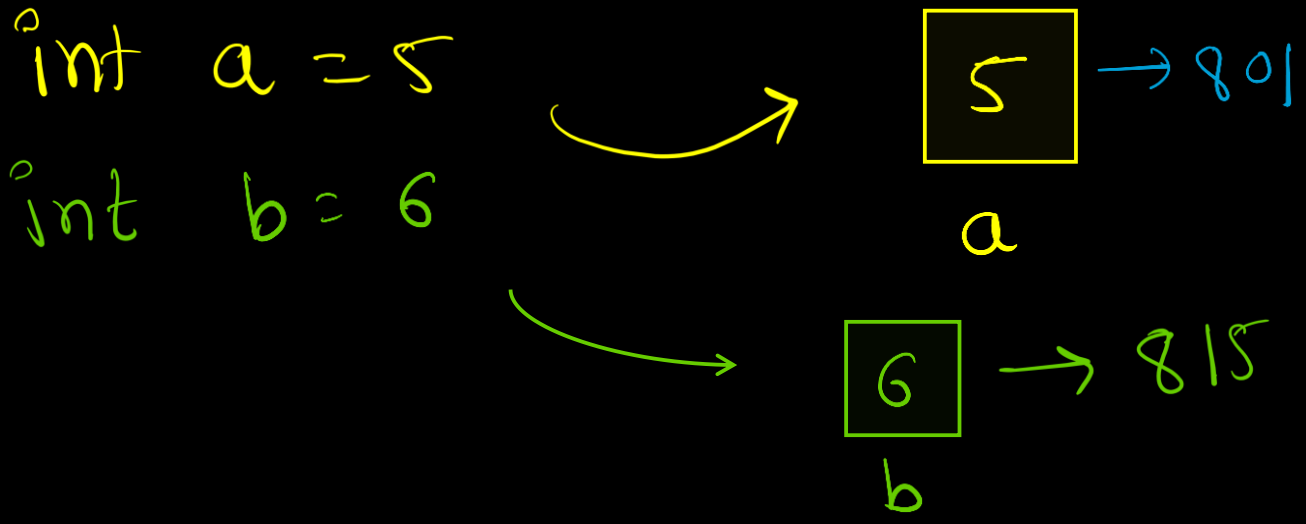
What is Array ?

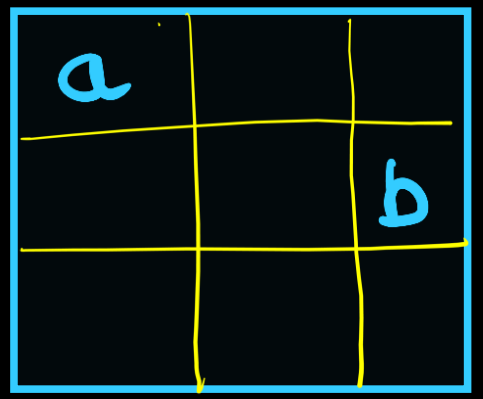
1. It is a linear data structure
2. Array is collection of same data type

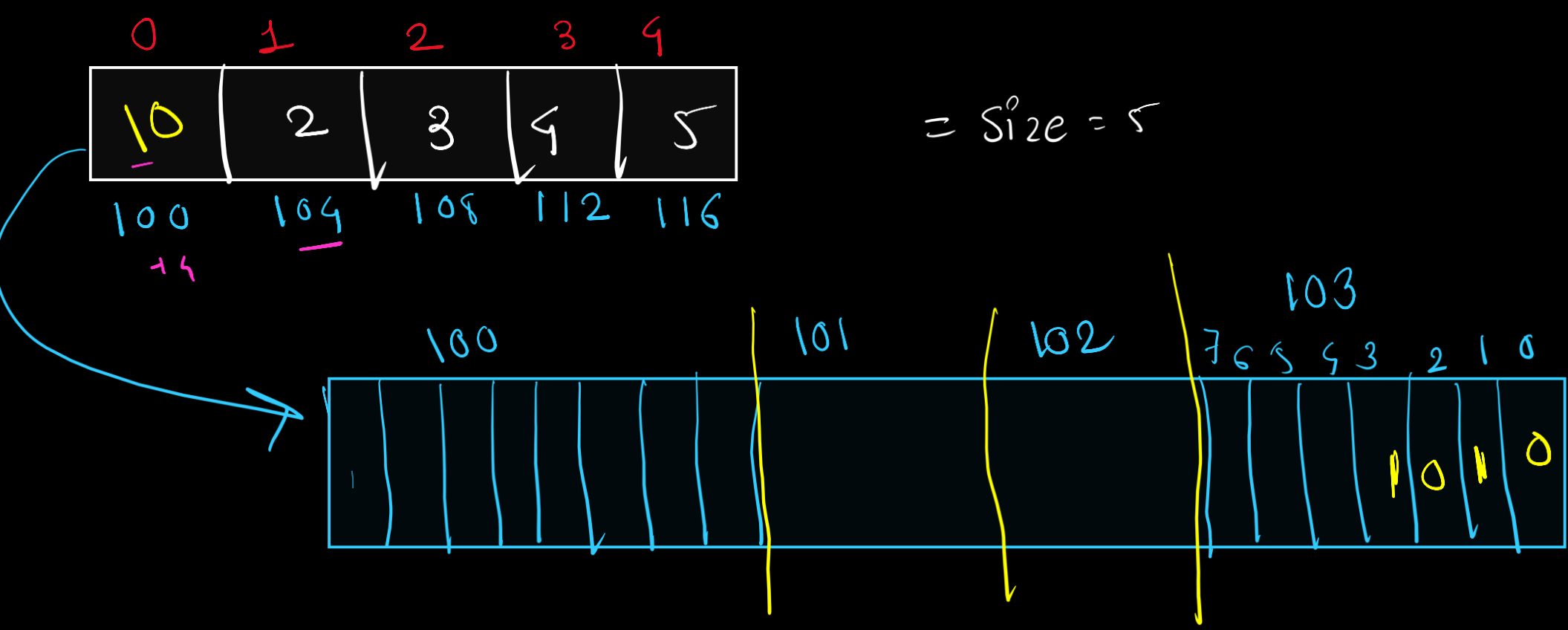
a. Either whole array fill with integer , float, or character we can't mix this data type in one array like -> arr = {1, 1.5, 0.5, 'a'} this is not possible in array

3. It store element in contiguous memory location
4. We can easily Access element in array



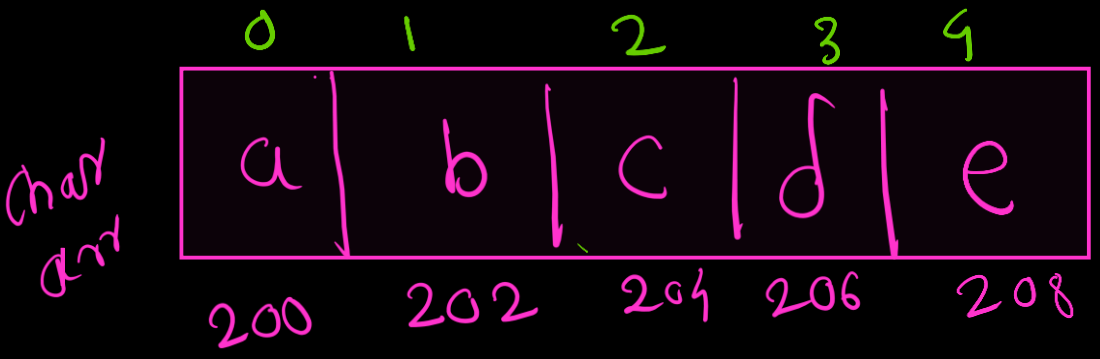






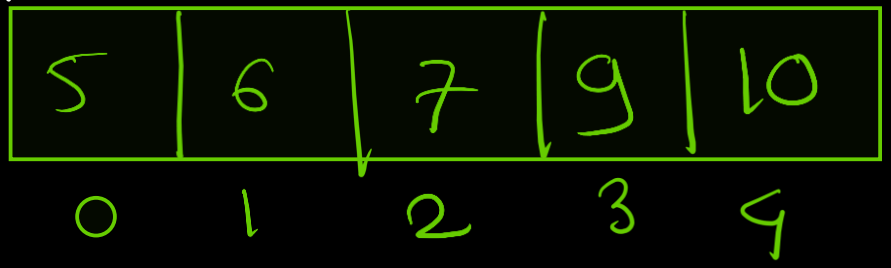
1 byte = 8 bits

4 -||- = 8 * 4 = 32



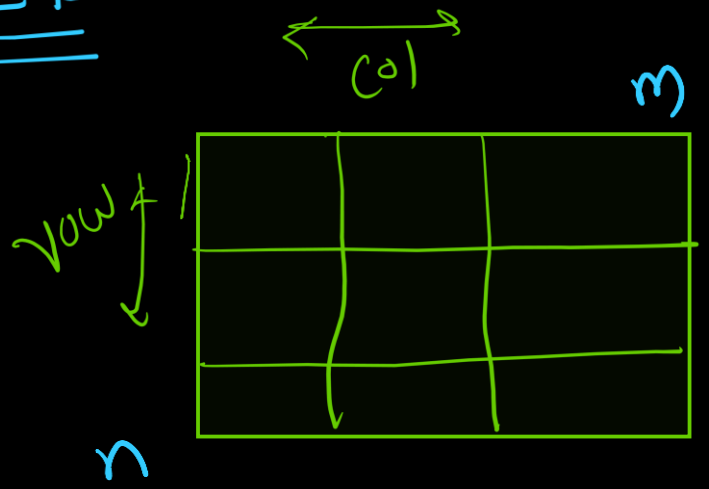
Types of Array

1. 1 D array

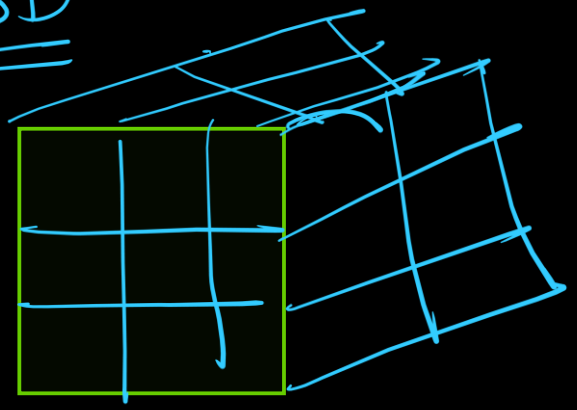


multidiamansional Array

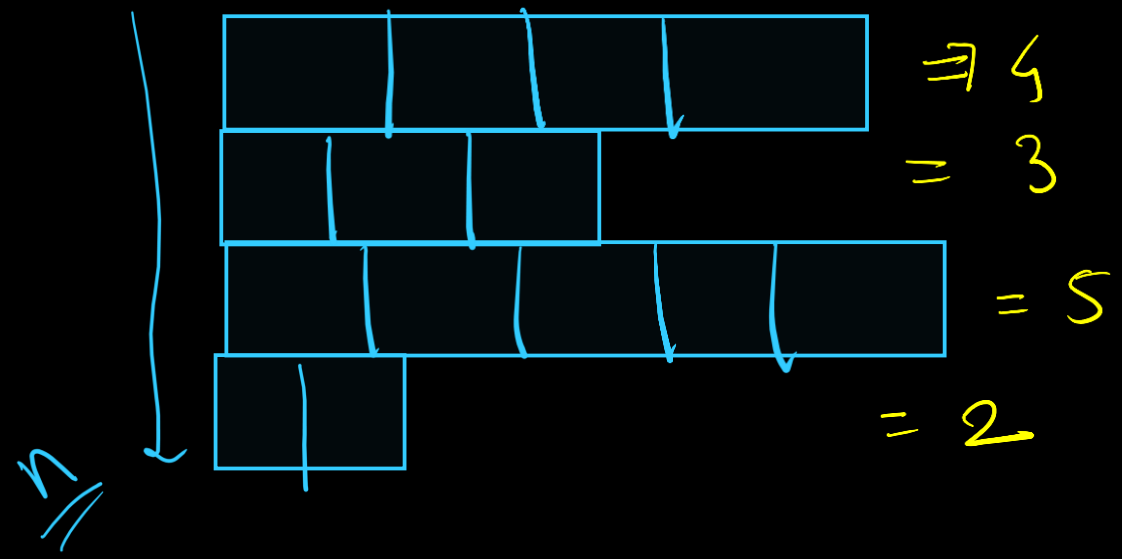
2D

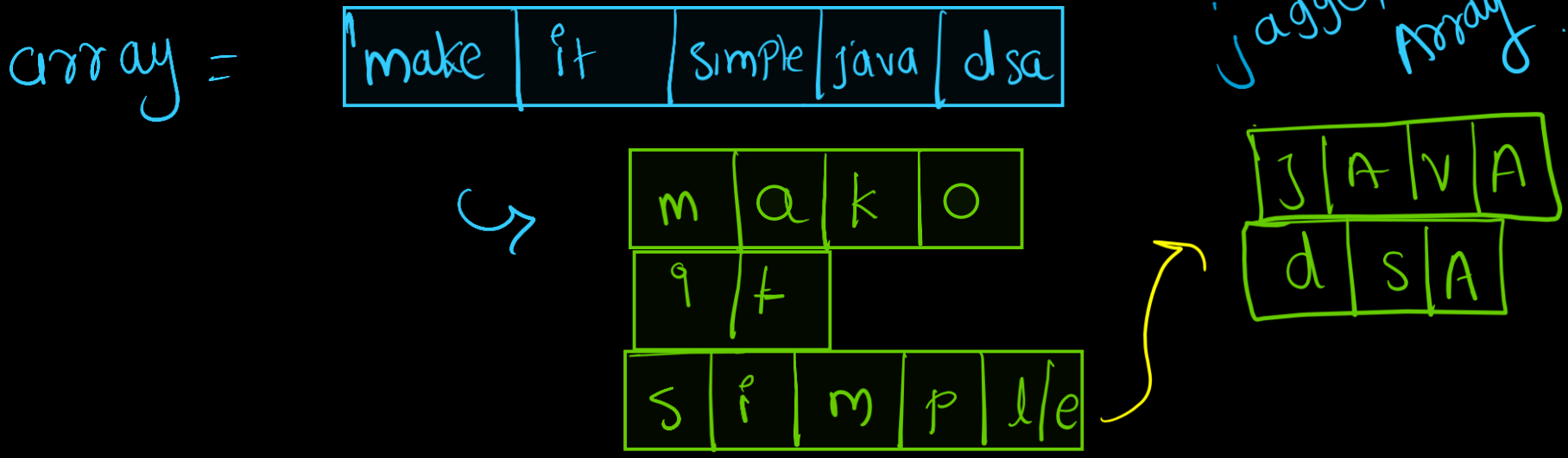


3D

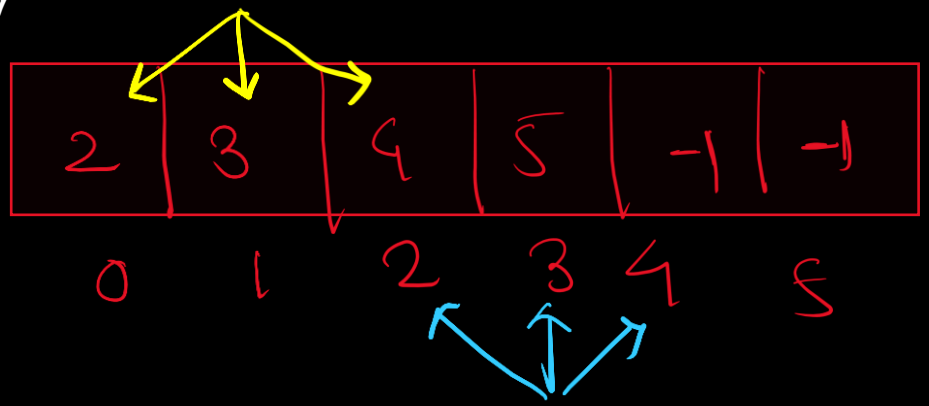


Jagged Array





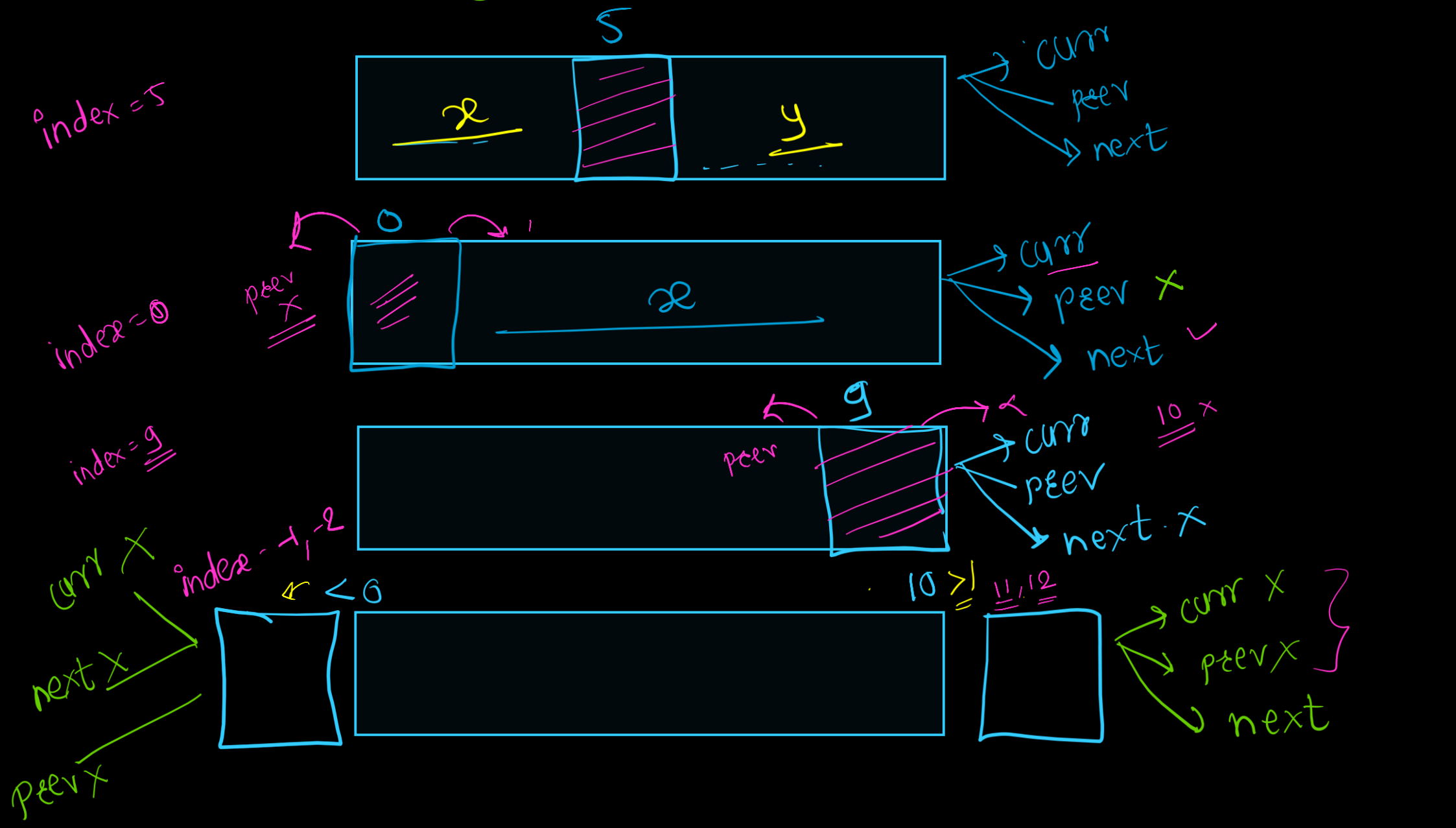
Insertion in Array

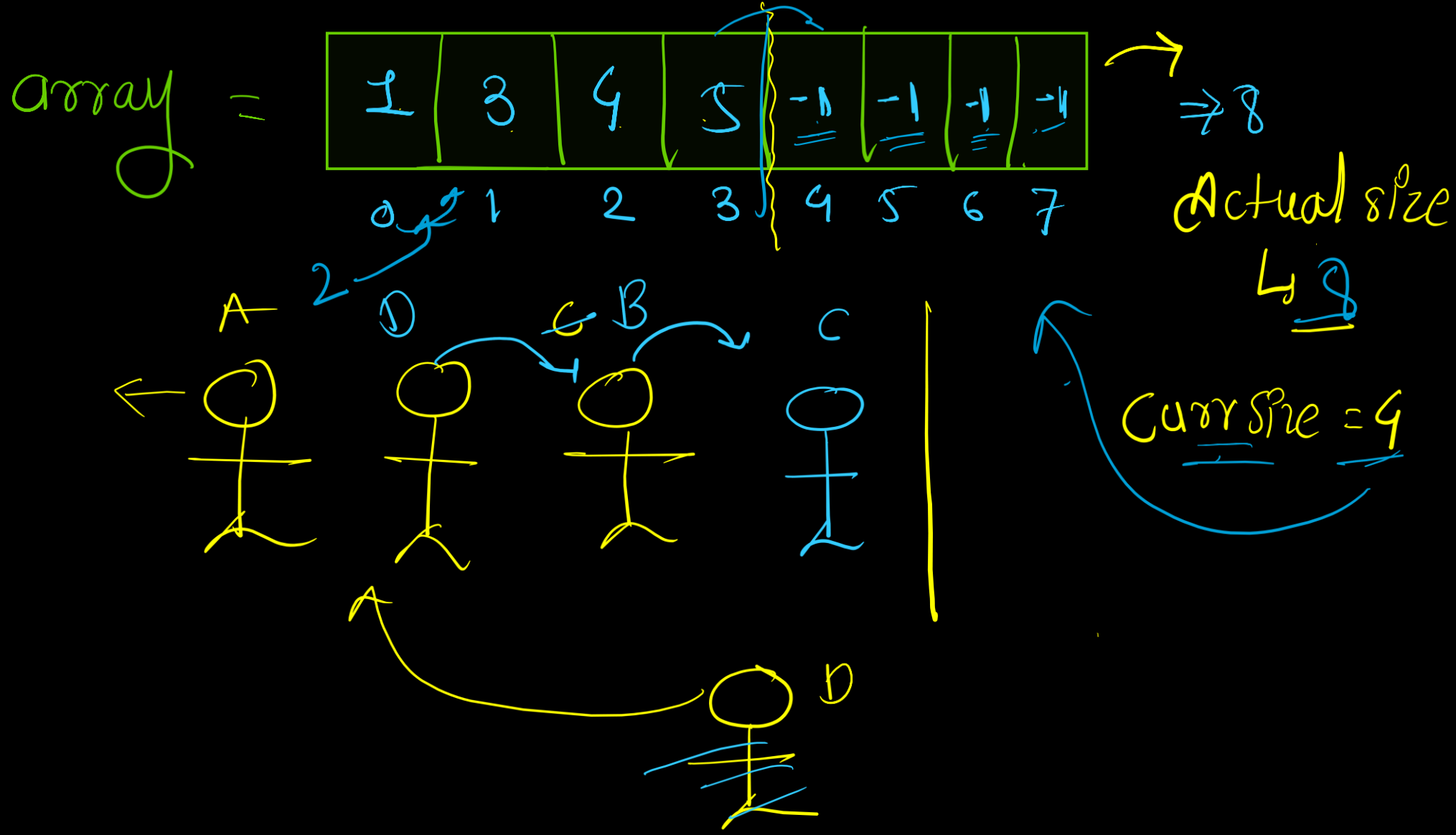


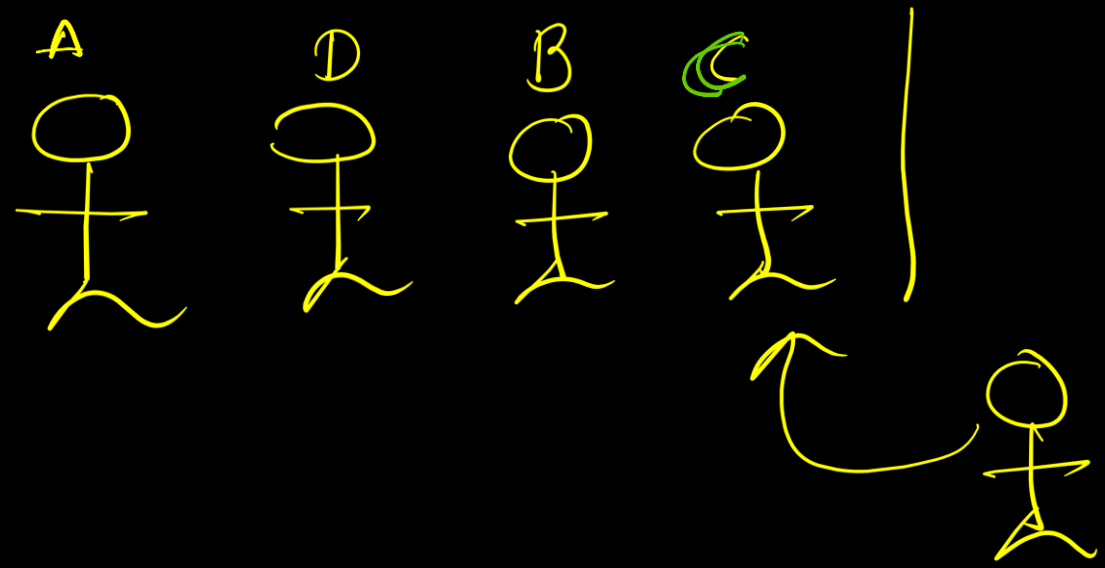
1) You have given a value.
   a) insert at that value
   b) insert after the value
   c) insert before the value

2) You have given an indexe.
   a) insert at that indexe.
   b) insert after the indexe.
   c) insert befor the indexe

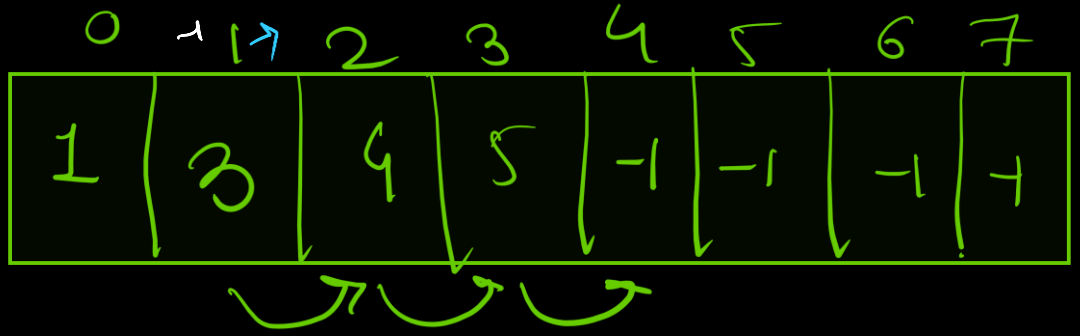
2) You have given an index.




array =
1
3
4
5
-1
-1
-1
-1
0
1
2
3
4
5
6
7
⇒ 8
Actual size
4 9
2
A
D
C
B
C
Curr Size = 4
A
D



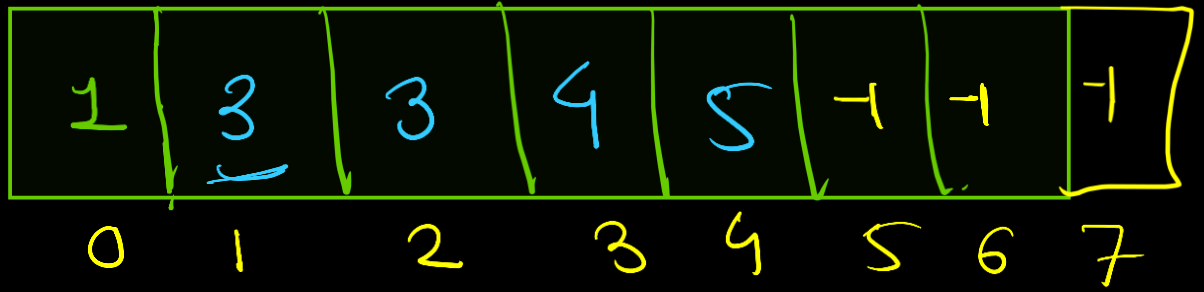

```
if (curSize < actualSize){
    ---- perform some operatn
}
else{
    We can't insert Array is full
}
```



actual size = 8

current size = 4

Insert 2 at index 1



Assign

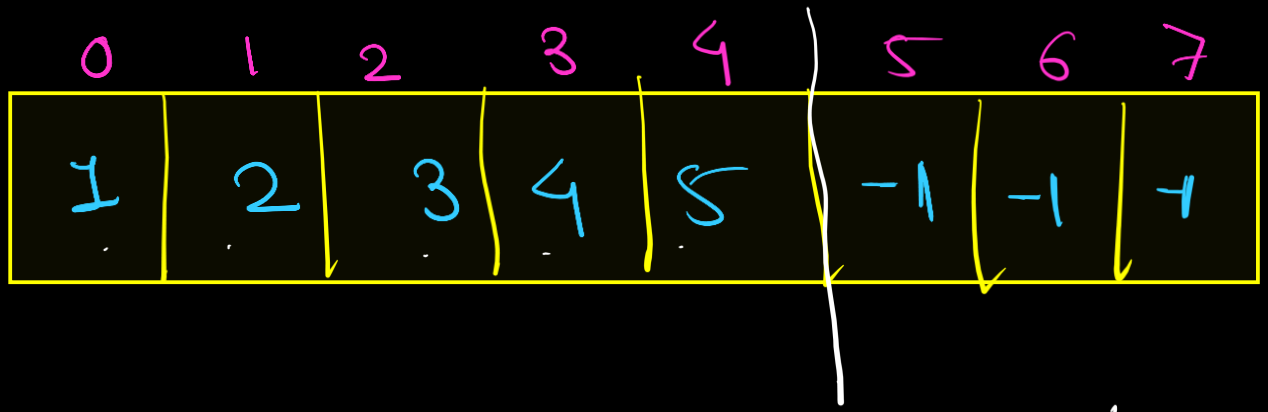
arr[1] → arr[1+1] → arr[1+1] = arr[1]

arr[2] → arr[2+1] → arr[2+1] = arr[2]

arr[3] → arr[3+1] → arr[3+1] = arr[3]

arr[index] = value

arr[1] = 2



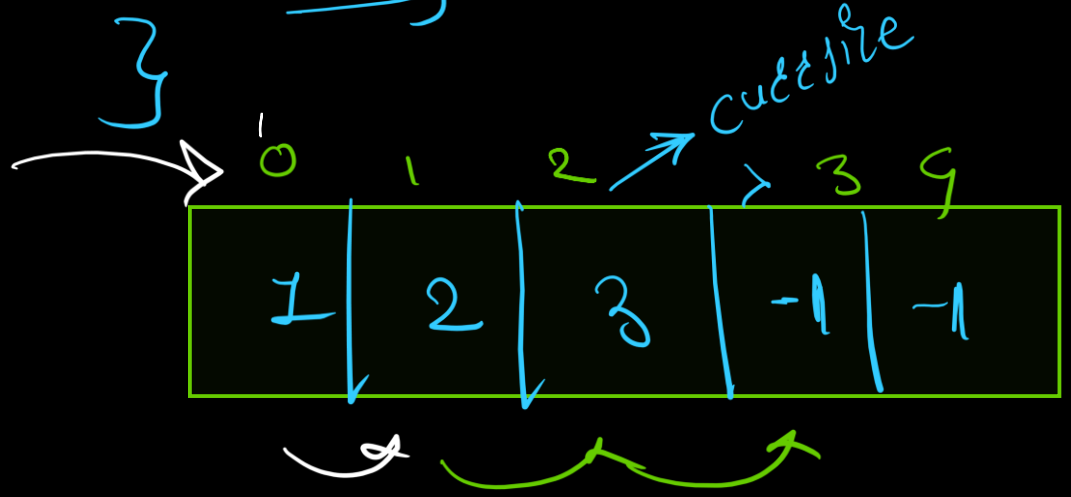
currSize = currSize +1 = 4 + 1 = 5

```
if ( currsize < actualsize ){
    if ( index < 0 || index >= actual size){
        print " index is invalid ";
    }
    else{
        //shift element
        for( i = currsize ; i >= index ; i--){
            arr[i+1] = arr[i];
        }
        arr[index] = value;
        curr Size = cursize + 1;
```

} }

else { → print Array is full;

}

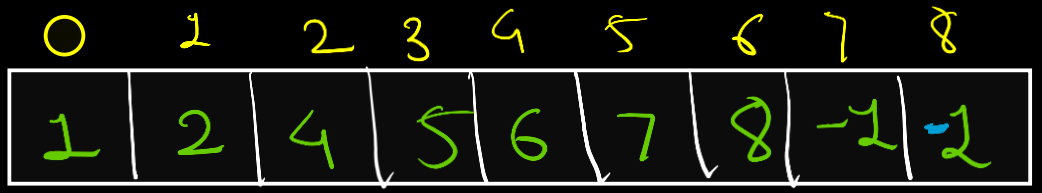



```
for( i = cursize; i >= index; i--){

    arr[i+1] = arr[i];

}
```

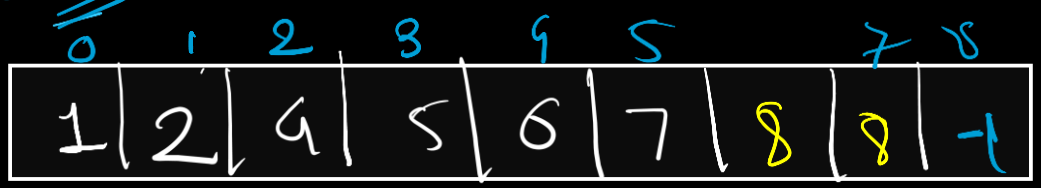
DRY RUN →



= Actual size = 9
currsize = 7.

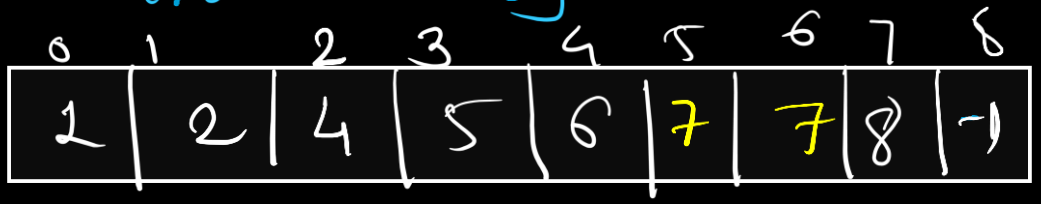
value = 8, index = 2

i = currsize = 6



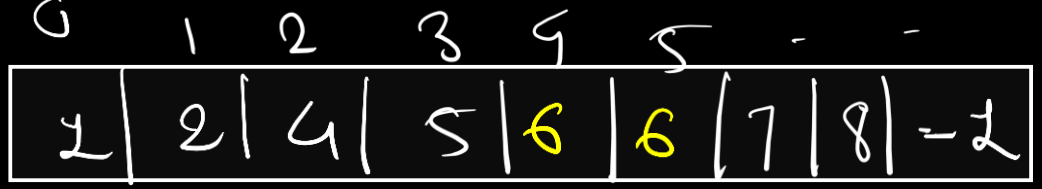
arr[7] = arr[6]

i = 5

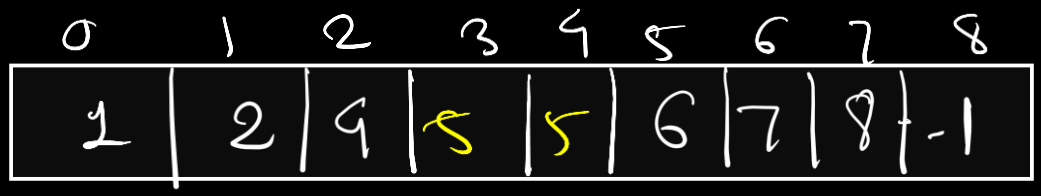


arr[6] = arr[5]

i = 4

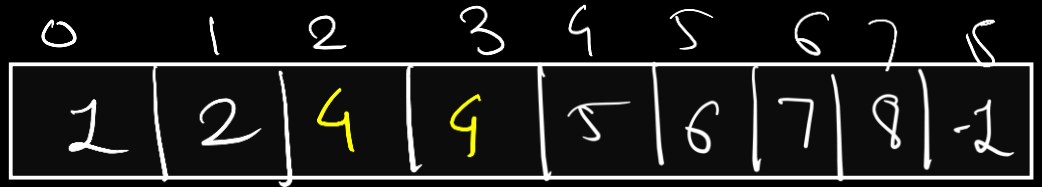


arr[5] = arr[4]



i=3

arr[4] = arr[3]

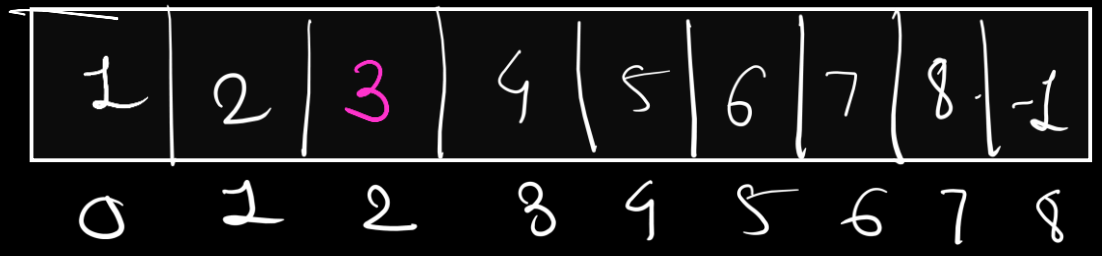


i=2.

arr[3] = arr[2]

End the loop =

arr[index] = value



CurrSize = CurrSize + 1 = CurrSize = 8