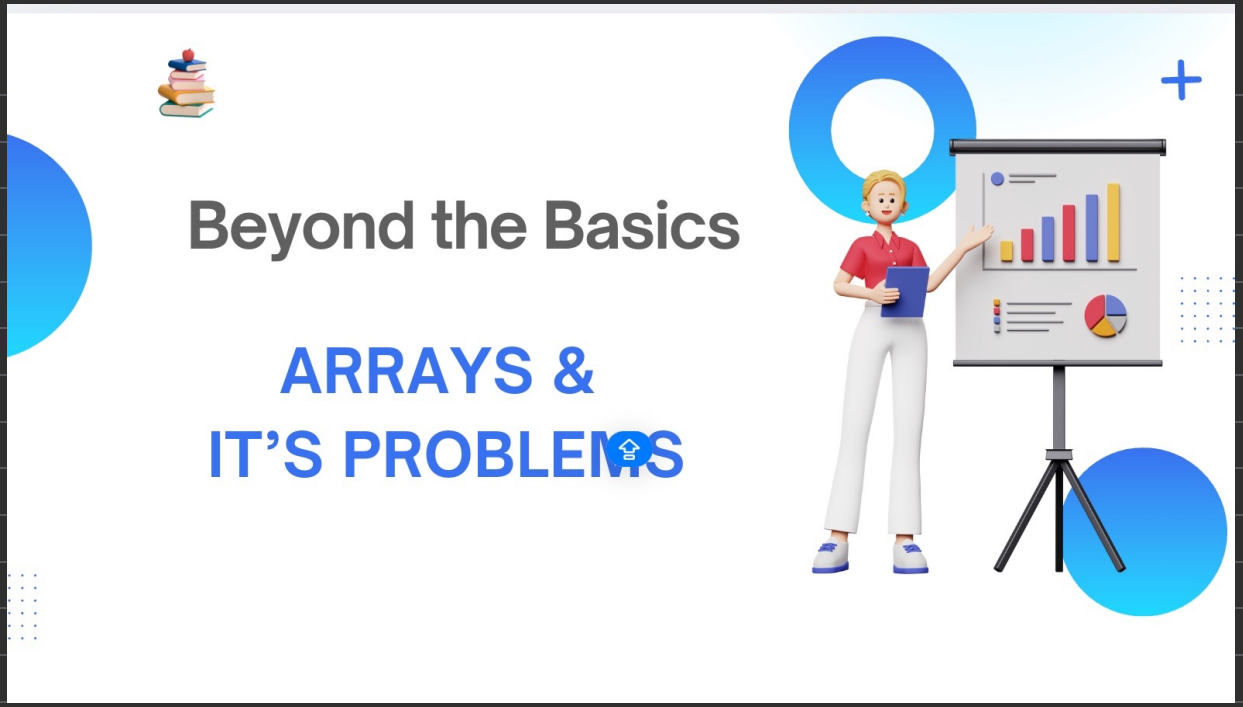


Always remember, beginning is the hardest part.

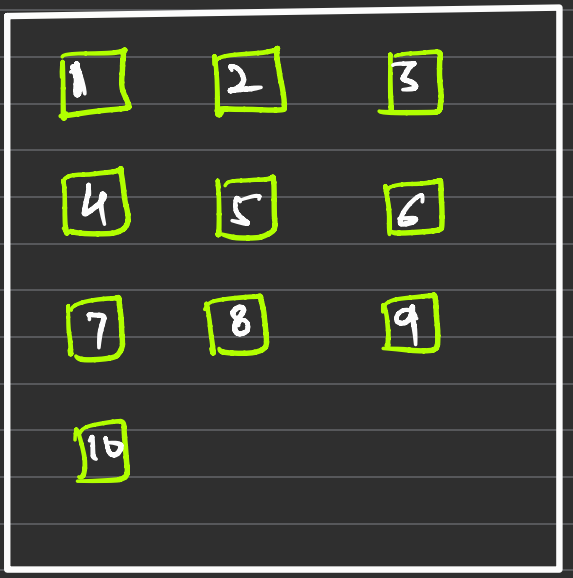
We go to the GYM because we are weak & we wanted to be strong. We go to Library, because we dont have knowledge & we wanted to gain some. Hence, process of discomfort is part of success.

Arrays:

- Why we need array
- What is array
- How array is Represent
- 3 / 4 problems

Why we need Array

Siddhu.



Class

$x_1 = 10$

$x_2 = 30$

$x_3 = 40$

$x_4 = 60$

$x_5 = 70$

$x_6 = 80$

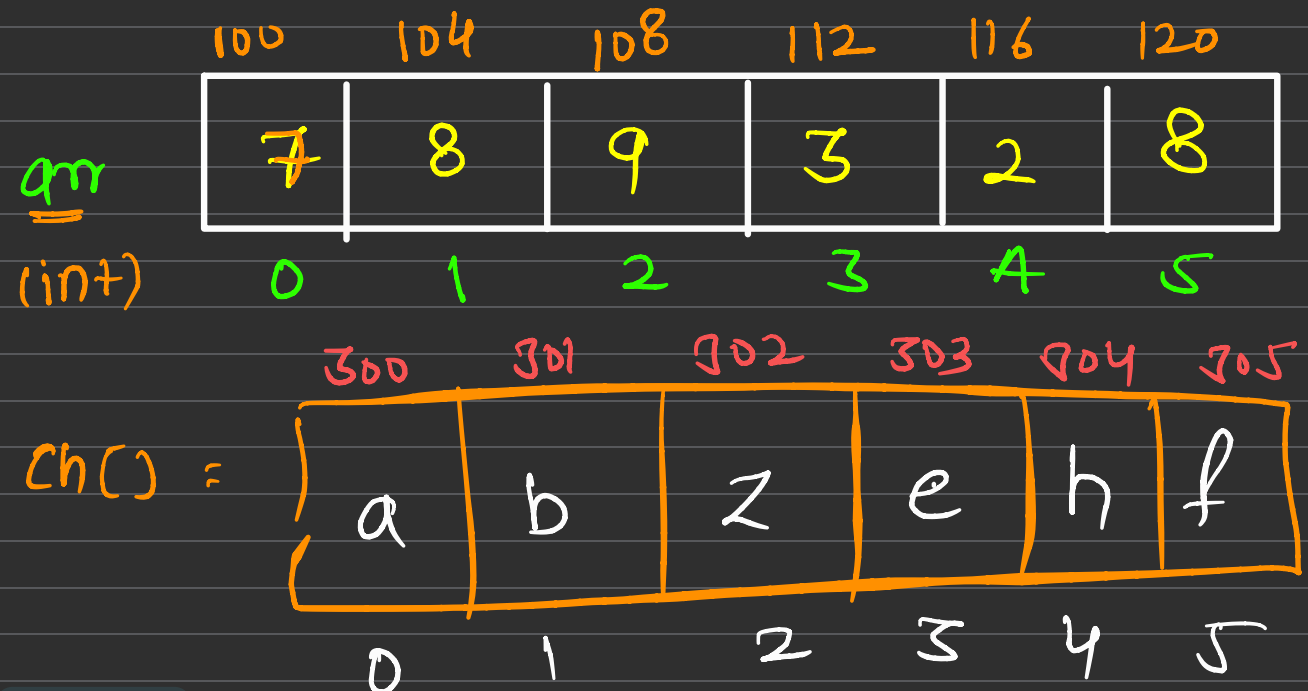
a(3)

a(5)

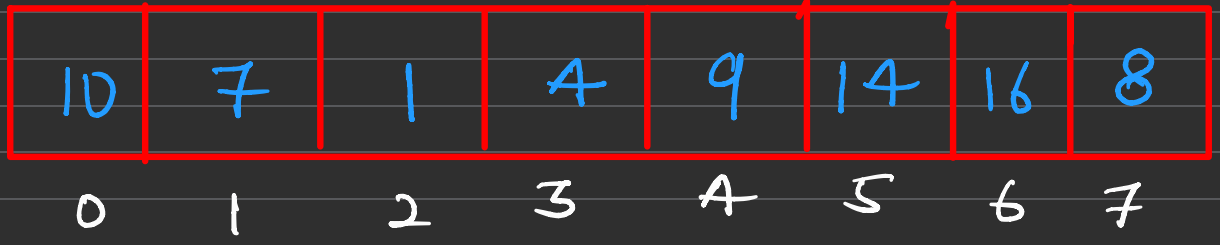
$$\underline{a} = \{ \underset{0}{x_1}\ \underset{1}{x_2}\ \underset{2}{x_3}\ \underset{3}{x_4}\ \underset{4}{x_5}\ \underset{5}{x_6} \}$$

- group of Element
- Similar Types

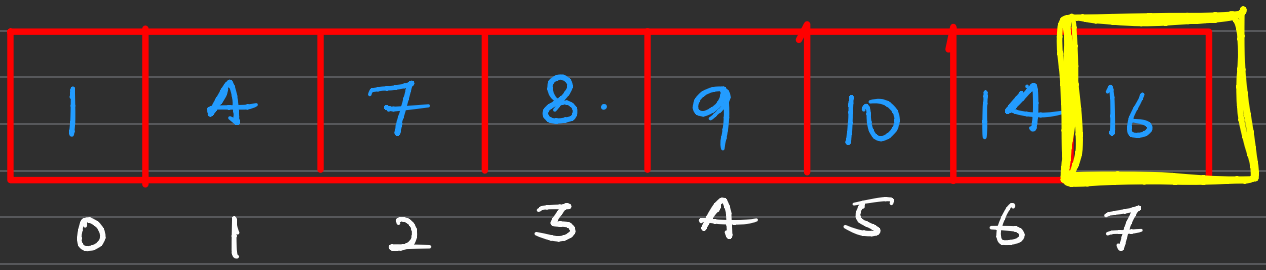
Array: Collection of elements of similar type contiguous memory allocation.

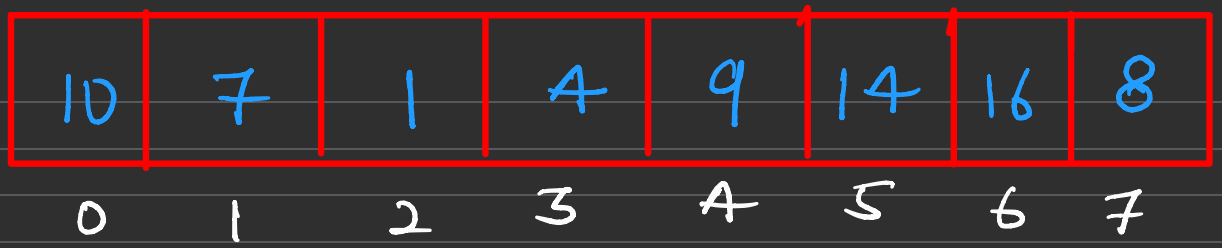


find largest Element from array ?



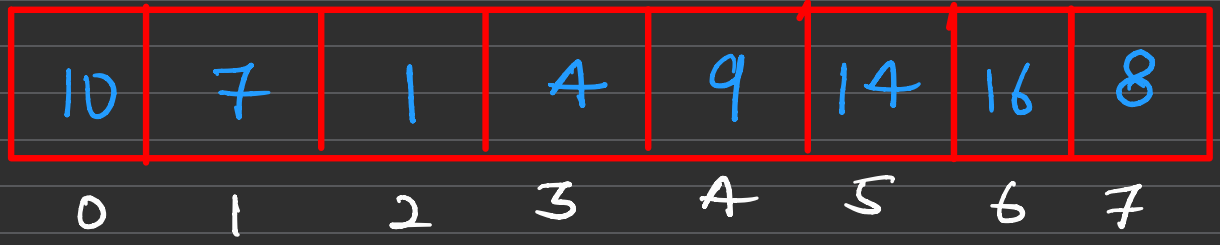
Sort: O(N.log N)

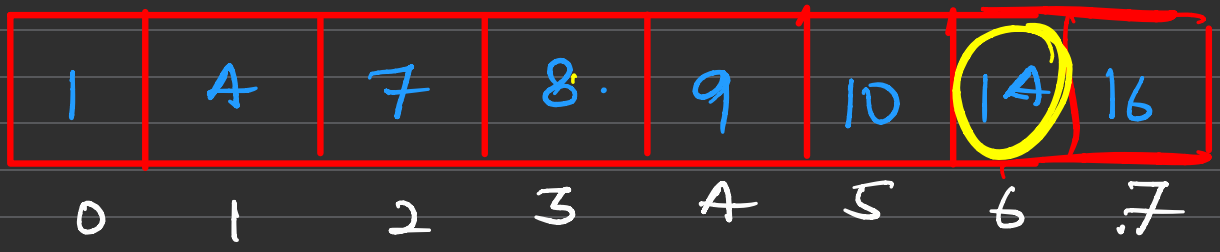


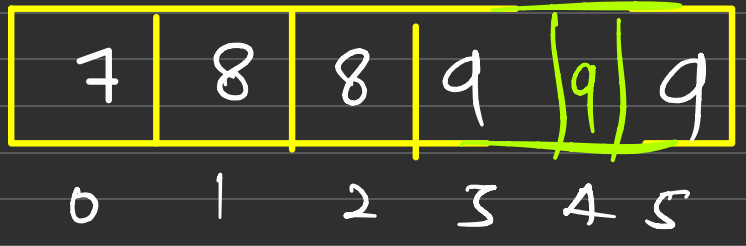



```
int temp = a(0)
for (i=0 ; i<n ; i++)
{     if (a(i) > temp)
      {   temp = a(i)
      }
}
(temp)
```

② find second largest element ?

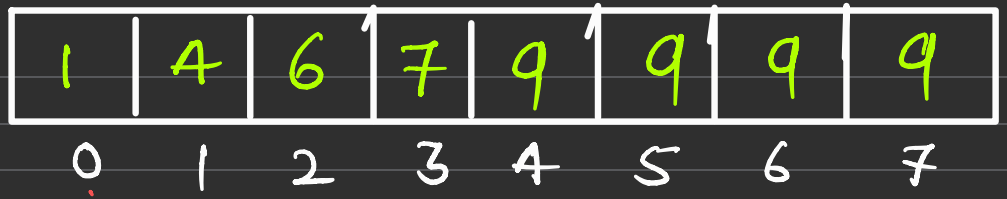




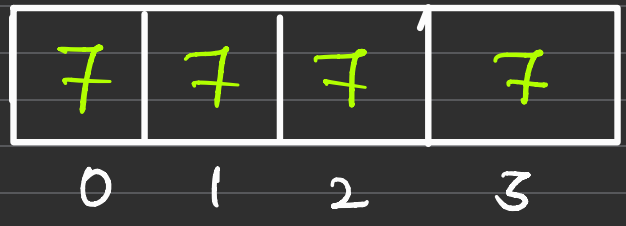


second largest

(-1)



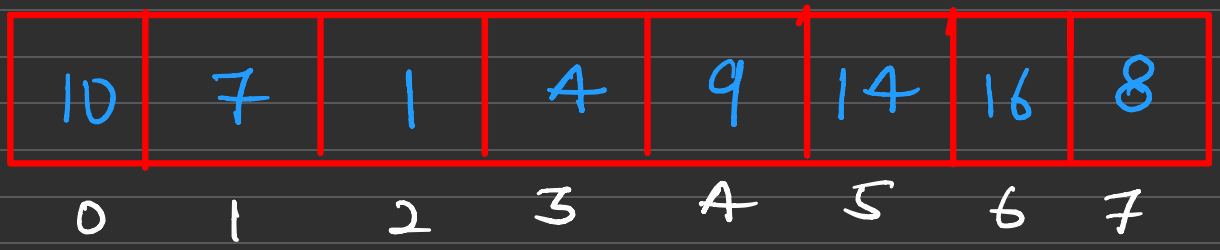

```
sort(arr)

for (i = n-2 ; i ≥ 0 ; i--)
{
    if (a[i] != a[i+1])
        return i / a[i]
}

return -1
```

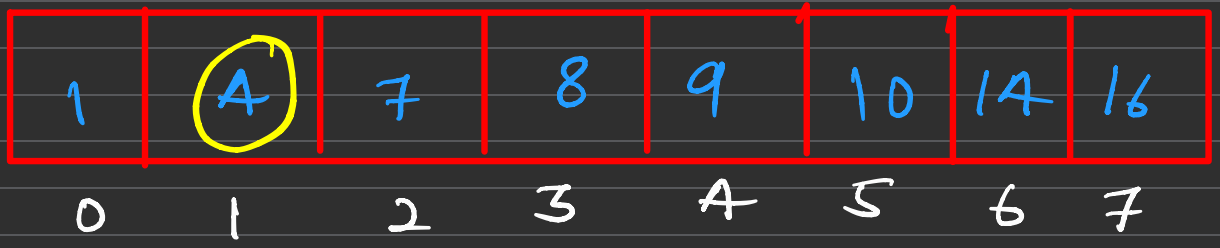


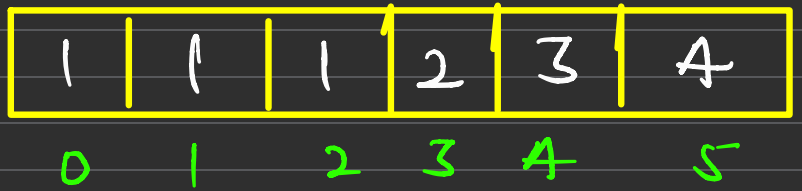
N = 4

i = ~~2 1 0~~ -1

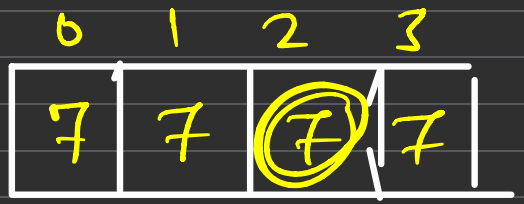
* Second Smallest








```
sort (arr)
for ( i=1 : i<n ; i++)
{
    if (a(i) != a(i-1))
        return a(i)
}
return -1
```



N = 4

i = 1 2 3 4