

Ques Double Occurrence →

arr1 =

1	2	3	4	5
---	---	---	---	---

 ✓
i

O/p = 1 2

arr2 =

1	2	1	2	3	4
---	---	---	---	---	---

 ✓
j

for (i = 0 to n) {
 count = 0;

 for (j = 0 to m) {
 if (arr1[i] == arr2[j]) {
 count++;

 if (count == 2) {
 syso (arr1[i]);
 }

```

public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr1 = new int[n];
    for(int i=0; i<arr1.length; i++){
        arr1[i] = scn.nextInt();
    }
    int m = scn.nextInt();
    int[] arr2 = new int[m];
    for(int i=0; i<arr2.length; i++){
        arr2[i] = scn.nextInt();
    }

    for(int i=0; i<arr1.length; i++){
        int count=0;
        for(int j=0; j<arr2.length; j++){
            if(arr1[i]==arr2[j]){
                count++;
            }
        }
        if(count == 2){
            System.out.print(arr1[i]+" ");
        }
    }
}

```

am1:-

1	2	3	4	5
0	1	2	3	4

am2:-

1	2	1	2	5	4	3	2	4
0	1	2	3	4	5	6	7	8

i = 0 + 2 3
count = 0 + 2

%p 1 3

Ques char, to int, int to char
string to integer

str.toInt()

n.toString()

String str = "27";

int n = Integer.parseInt(str);

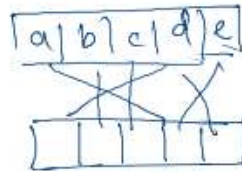
int n = 27

String str = "Kartik";

Time Complexity

→ time - How much time my code is taking to be executed
→ space → How much memory is utilized for this execution.

Ques → Array :→ Merge Sort

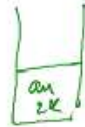


extra space

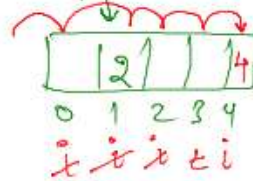
arr[1] = 2;

```
for(i = 0; i < arr.length; i++)  
{  
    if(i == 4){  
        arr[i] = 4;  
    }  
}
```

↓
dependent on # of element
n-element
↳ $O(n)$



heap



$O(1) \rightarrow \text{constant}$

$O(1) \times n$
 $= O(n)$

$$1 \rightarrow 10^{18} \text{ ms} \\ = 2 \text{ sec}$$

$$1 \rightarrow 10^9 \text{ ms} \\ = 1 \text{ sec}$$

$$n = \frac{2^{31} - 1}{2} \Rightarrow \boxed{15\text{ sec}} = O(n)$$

$$\rightarrow 2^{31} = 2^{30} \times 2 \\ \approx 2^{30}$$

↓

$$\left(\frac{10}{2}\right)^3 = (1024)^3 \\ \approx (10^3)^3 = 10^9 \rightarrow 1 \text{ sec}$$

Ans
m
h.
f. } Optimized Solution

↳ Brute force $O(n^2)$

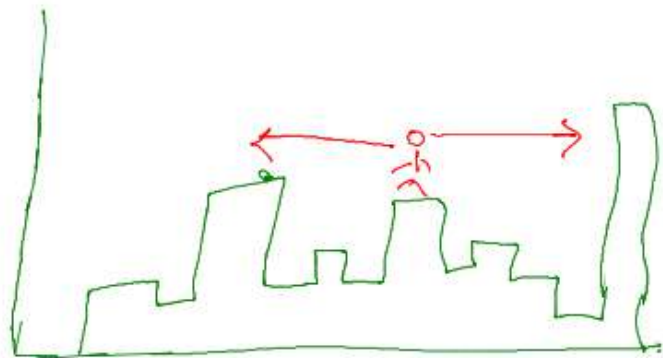
↳ Optimized $\rightarrow O(n)$

$O(2)$

↳ $O(\log n)$

Ex trapping Rainwater

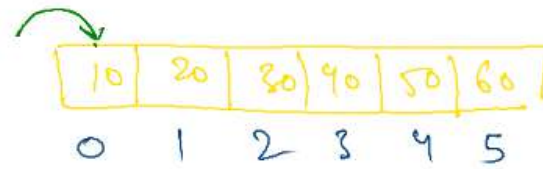
→



$$\rightarrow \frac{n}{2} \times \frac{n}{2} \times 2 \\ = n^2 \rightarrow \text{Brute force} \rightarrow \boxed{n} \checkmark$$

Linear Search

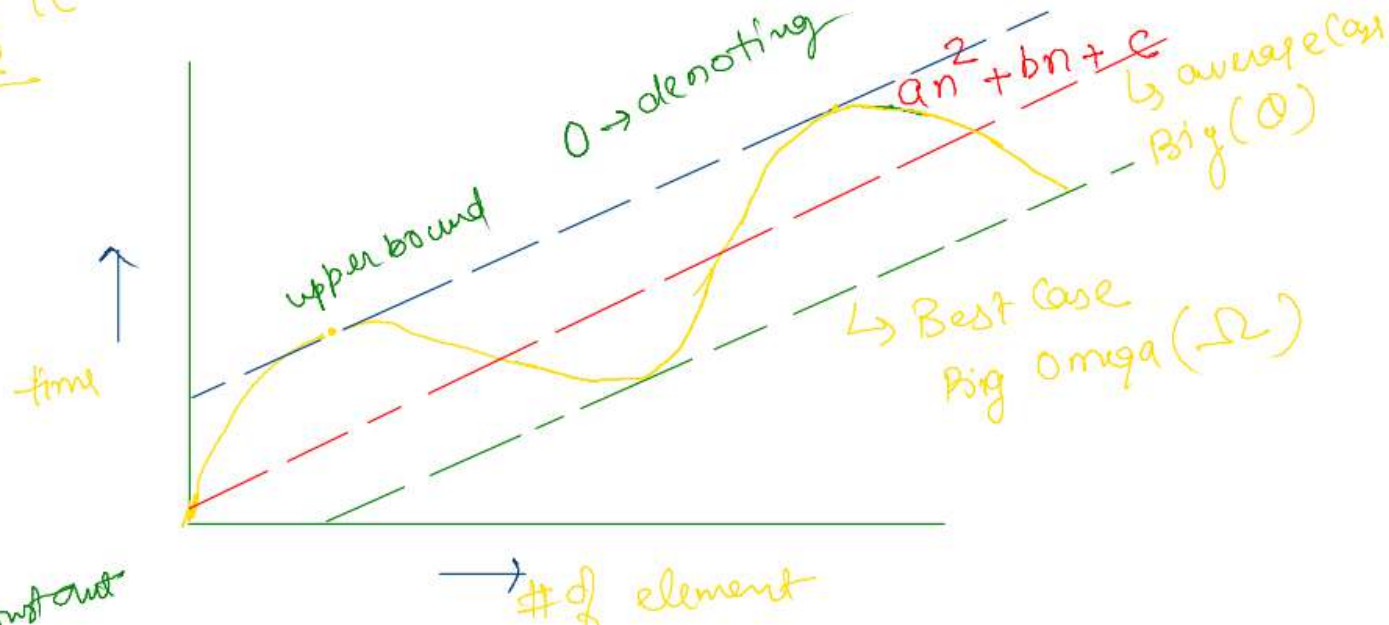
let arr length = n



target = 40

41	—	worst case $\rightarrow n$
40	—	Average Case $\rightarrow \approx \frac{n+1}{2}$
10	—	Best Case $\rightarrow 1$ sec

Type of TC



let $\text{Big}(O)$ constant

\checkmark $\left[\begin{matrix} 10 \times 10 \times 2 \\ n \end{matrix} \right] = O(n^2)$
 24 right constant

\checkmark $a(n^2) + b(n) + c$ \rightarrow # of element
 \checkmark \rightarrow addition
 \downarrow inner loops (two loops) \downarrow one loop
 when $\{a, b, c \text{ are constant}\}$
 $= O(n^2)$

for loop
 for
 addition

\checkmark \checkmark

$arr1 = \begin{matrix} i & i & i & i & i & i \\ \boxed{1} & \boxed{2} & \boxed{3} & \boxed{4} & \boxed{5} & \boxed{6} \end{matrix} = n \text{ length}$
 $0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5$
 $= n \times n = n^2 = O(n^2)$

$arr2 = \begin{matrix} \boxed{1} & \boxed{1} & \boxed{2} & \boxed{2} & \boxed{2} \end{matrix} = m \text{ length}$
 $0 \quad 1 \quad 2 \quad 3 \quad 4$
 $\Rightarrow O(n)$

\checkmark for ($i \rightarrow 0$ to n) {
 $\equiv O(1)$
 for ($j = 0$ to m) {
 $\equiv O(1)$
 \downarrow
 $\equiv O(1)$
 o/p
 $\Rightarrow O(n^2)$

$0 - m$
 $1 - m$
 $2 - m$
 \vdots
 $m \text{ times}$
 \vdots
 $n - m$

elements = $(O(n \times m), O(n))$
 $= O(n \times m)$

$\text{int } [] \text{ arr} = \text{new int } [n];$
 for $\{ i=0 \text{ to } \text{length} \}$ $\{ O(n)$
 $\{$
 $\text{for } (i=0 \text{ to } \text{length}) \{$
 sum(arr[i]); $\} O(n)$
 $\{$
 $\}$
 $\}$
 $O(n) + O(n)$
 $2 \times O(n)$
 $= O(n)$

Q → 1st student $i \quad i \quad i \quad \dots \rightarrow$

1	2	3	4	5	6
0	1	2	3	4	5

 $n=6$

$$\begin{aligned}
 0 \rightarrow n &= n + n-1 + n-2 + \dots + 1 + 0 \\
 1 \rightarrow n-1 &= 6 + 5 + 4 + \dots + 1 + 0 \\
 2 \rightarrow n-2 &= 6 + 5 + 4 + \dots + 1 + 0 \\
 &\vdots \\
 n \rightarrow n-n=0 &\Rightarrow \frac{n(n+1)}{2} \Rightarrow \frac{1}{2}(n^2+n) \\
 &= O(n^2+n) = O(n^2) \checkmark
 \end{aligned}$$

2nd student $i \quad i \quad i \quad i \quad i \quad i \quad i \quad i \leftarrow$

1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7

 right $O(n)$ \rightarrow right max \rightarrow

15	15	15	15	0	8	8	0
0	1	2	3	4	5	6	7

$\text{max} = 8, 15$
 $\Rightarrow O(n)$

left $O(n)$

$O(n)$ $\{$ for $(i=0 \text{ to } \text{length}) \{$
 $\text{ans} += \text{max}(\text{right}[i], \text{left}[i])$
 $= \text{arr}[i]$
 $\{$
 sum(ans)

$TC = O(n) + O(n) + O(n)$
 $\Rightarrow 3 \times O(n) = O(n)$