Time & Space Complexity

Highest No.

arr → [1, 5, 10, 12, 5]   Find dublicate

① Using double loop. TC n^2 SC = n(1)

~~②~~ Using loop   Arr2 :

0 1 2 3 4 5 6 - -
10 11 12

ⓐ Its like, create an array of size (X) whatever mention in question.

ⓑ Using for loop

```
for(int i = 0; i < len.of.Arr; i++){
   // check if Arr2 of index arr of i 0 or 1
   // If 1 it mean that element is
   //   already in Array
   //   else go to that index make it 1,
if (arr2[arr[i]] = 1){ here is your target}
else{ arr2[arr[i]] = 1}
```

T.C is O(n) But S C may be worst ✓

I don't think this is correct

③ Iterate the array and sum all the elements.

Then, use this formula $\boxed{\frac{n(n+2)}{2}}$

It like arr = [1, 2, 5, 10] → sum = 18

using formula = $\frac{4 \times 5}{2} = 10$

But there is no element which occur twice

So, I can say this is only valid for whole no.

like Arr = [1, 2, 3, 4] = 10

using formula = $\frac{4 \times 5}{2} = 10$ ✓

Notations for different types of TC

1. Big oh ---> upper bound
2. Big omega ----> lower bound
3. Theta ---- > may be Avg

Q. Calculate TC for this loop

```
for(int i = 0; i < n; i++){
        cout << "Devil";
}
```

Here T.C is $O(n)$

```
for(int i = 0; i < n; i+=2){
        cout << "Devil";
}
```

still T.C is $O(n)$

All Are Same

$O(\frac{n}{2}) \simeq O(n)$

$O(2n) \simeq O(n)$

$O(n+k) \simeq O(n)$

$O(5n+K) \simeq O(n)$

But

T.N.O $\Rightarrow 5n^3 + 3n + 2$

$\Rightarrow O(n^3)$

Q. 2

```
for ( int i = 1; i < n ; i *= k ){
        cout << "hello";
}
```

output => if n 100, and k = 2
==> hello only 7

i = 1 hello
i = 2 "
i = 4 "
i = 8 "
i = 16 "
32 "
64 "
128 > n

TC $\Rightarrow O(\log_k n)$

$= O\left(\frac{\log n}{\log k}\right)$

$\Rightarrow O(\log n)$

Q3.

```
c = 1;
for( int i = 0; i < n; i++){
        for(int j = i+1; j < m; j++){
                c++;
        }
}
```

$i = 0$

$j = 1, 2, 3 \dots \dots m-1$

$i = 1$

$j = 2, \dots \dots m-1$

$\vdots$

$i = n$

$j : n, n+1, \dots m-2$

$$TC = \underbrace{1+2+\dots+n}_{\text{for 1st loop}} + (m-1) + (m-2) + (m-3) \dots \dots + (m-n)$$

$$\Rightarrow \frac{n(n+1)}{2} + (m+m+m+\dots)\,\text{ntimes} - \left(\frac{n(n+1)}{2}\right)$$

$$\Rightarrow O(mn)$$