Date _____ Page _____

Day -38

Two Pointer in C++

* **Segregate 0 & 1:**

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

⇒ We have put all 0 in the starting then 1.

⇒ We can also do this by sorting the array. → $O(N^2)$

⇒ We can also sort it by ~~any~~ built-in sort function of vector. → $O(N \log N)$

⇒ We can do this question by counting the no. of 0 & 1 then insert the no. of 0 in the starting & after that 1.

Code:

```
int count0 = 0;
int count1 = 0;
for( i=0; i<n; i++){
    if(arr[i] == 0)
        count0++;
    else
        count1++;
for(i=0; i< count0; i++)
    arr[i] =0;
for(i= count0; i< n; i++)
    arr[i] = 1;
```

n

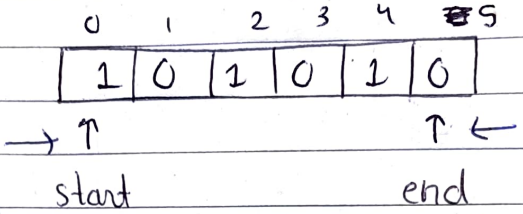
n

n

⇒ T.C → O(n) , S.C → O(1)

Date _____ Page _____

⇒ So, can we do the same task in one traversing. For this, two pointers ~~can comes~~ help us to do this.



⇒ So, we will check start & end values if they are in correct order then we do nothing otherwise ~~to~~ swap the values. and increase the pointer of start & decrement in end.

⇒ when start crosses the end then stop the process.

Code:

```
int start = 0, end = n-1;
while (start < end ){
    if (arr[start ] == 0 ) {
        start ++ ;
    else {
        if (arr[end ] == 0 ){
            swap (arr[start], arr[end ]);
            start++ ; end-- ; }
        else
            end-- ;
    }
}
```

Date _____ Page _____

* Two Sum:

2	7	11	15	27

Target = 22

⇒ We have to find two number that sum equals to the target.
Just like, 7 + 15 = 22
Here, ~~two~~ those two no. are 7 & 15.

⇒ In Brute Force Approach, we will check one by one with every element and when we will get our answer return the index of both the no..

⇒ In Second Approach, we use binary search. we will start with first no. then to find second no. we use binary search.

⇒ In third approach, we use two pointers start and end.

0	1	2	3	4
2	7	11	15	27

Target = 22

↑ start →

← ↑ end

=⇒ Start always increase the value.
=⇒ End always decrease the value.
=⇒ So, if you want to increase the value then increase the start and if you want to ↓ the value ↓ the end.

Date Page

⇒ 2+27 = 29 > target
⇒ Decrease the end.
⇒ 2+15 = 17 < target
Increase the start
⇒ 7+15 = 22 = target.

* Pair with Given Difference

| 5 | 10 | 3 | 2 | 50 | 80 |

Diff = 45

⇒ We have to find two no. that their diff. is equal to the required difference.
⇒ In First Approach, you can iterate whole array & find the no. ~~that~~
⇒ This is our Brute Force Approach.

⇒ In second Approach, we can use BS.
⇒ As we do in last question, same we can do in this question.

⇒ In Third Approach, we can use two Pointers.
⇒ First, we sort the array.
⇒ Then, we take two pointers – start & end.

| 2 | 3 | 5 | 10 | 50 | 80 |

Diff = 45

↑ start ↑ end

⇒ if end - start < diff
⇒ ↑ end.

Date ________ Page ________

if end-start > diff
↑ start.