

# \* Cyclic sort \*

Imp

\* When gives nos. from range  $1, N \Rightarrow$  Use cyclic sort.

\* what is cyclic sort & how it works

e.g.

3, 5, 2, 1, 4

when the array is sorted in that case all the numbers are going to be at their correct index.

	0	1	2	3	4
After sorting	1	2	3	4	5

{ Here after sorting index will be come value-1 }

Index = value - 1 { using this we are going to sort array }

why?

Because index starts from zero.

check-swap - Move

## \* Worst Case

3, 5, 2, 1, 4

check if 3 is at the correct index or not, if not do  $3-1=2$  (index = value - 1) then swap with correct index.

2, 5, 3, 1, 4



after swapping we know that 3 is at correct position now, but we do not know whether the other number that come at the position of 3 is correct or not so check again

0 1 2 3 4  
 2, 5, 3, 1, 4  
 swap

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

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

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

Now we'll check if it is at the correct position, if it is then move forward & so on.

\* we know that every unique item is only getting swapped once.

\* Hence we are not incrementing  $i$  when we are swapping so that might result in more than ' $n$ ' iteration of the loop.

\* worst case :-  $N-1$  (swap)

$$(N-1) + N$$

$$(2N-1)$$

$$\therefore O(n) \quad \text{linear}$$

```
Static void CycleSort(int[] arr)
{
    int i = 0;
    while (i < arr.length) {
        int correct = arr[i] - 1;
        if (arr[i] != correct arr[correct]) {
            swap(arr, i, correct);
        }
        else {
            i++;
        }
    }
}
```