Cyclic Sort :-

When given numbers from range 1 to N → use cyclic sort

Eg:

0	1	2	3	4
3	5	2	1	4

Q. What is cyclic sort and how it works ?

3, 5, 2, 1, 4 → { Here the no.'s are jumbled. but are from 1 to 5 }

let say N = 5

{ When the array is sorted in that case all the numbers are going to be at their correct ~~index~~ indices }

0	1	2	3	4
1	2	3	4	5

so, after Sorting = 1, 2, 3, 4, 5 - { Here after sorting index will become value - 1 }

index = value - 1 - { using this we gonna sort the array }

Why ?

Ans Because index starts from 0

CHEK - SWAP - MOVE

Q. might be :- you have given the array find the missing number ?

or you've given the no. from 1 to n find the duplicate no?

* worst case

Eg: 3, 5, 2, 1, 4 (0 1 2 3 4)

(1) - § Check is 3 at the correct index if not do 3-1 = 2 (index = value - 1) the 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 came at the position of 3 is correct or not so, check again §

(2) § If 2 is at correct position no it's not the again swap it, because it should be at index no 1. but it's at no. 0. so swap it with index 1 §

2, 5, 3, 1, 4

(3) 5, 2, 3, 1, 4

- § same theory as above §

(4) 4, 2, 3, 1, 5

- § same §

(5) 1, 2, 3, 4, 5

§ Now, we'll check if 1 is at the correct position, if it is then move forward. and so on § hence, it'll be our ans.

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

* Here we are not incrementing i when we are swapping so that might result in more than 'n' iterations of the loop.

* Worst Case = N-1 (swap)
(N-1) + N
= (2N-1)
∴ O(n) linear.

```
public class CyclicSort {
    public static void main (String[] args) {
        int[] arr = {3, 5, 2, 1, 4};
        sort(arr);
        System.out.println(Arrays.toString(arr
    }

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

    static void swap (int[] arr, int first, int secon
        int temp = arr[first];
        arr[first] = arr[second];
        arr[second] = temp;
    }
}
```