★ When given nos from Range [1,N] / [0,N] use Cyclic Sort

CYCLIC SORT

- Cyclic sort is an algorithm that works particularly well when you have an array of integers in range [1,N]
- It's an in place sorting algorithm that takes the advantage of the fact that each element in the array can be placed in its correct position by performing a cyclic permutations of elements.

Working-

Check : Start by comparing each number with its intended position

Swap : If a number isn't where it should be swap it with the number at that intended position.

Repeat: Continue this process for the swapped number until its in its correct spot

Move : Proceed to the next unprocessed number and repeat the check and swap steps

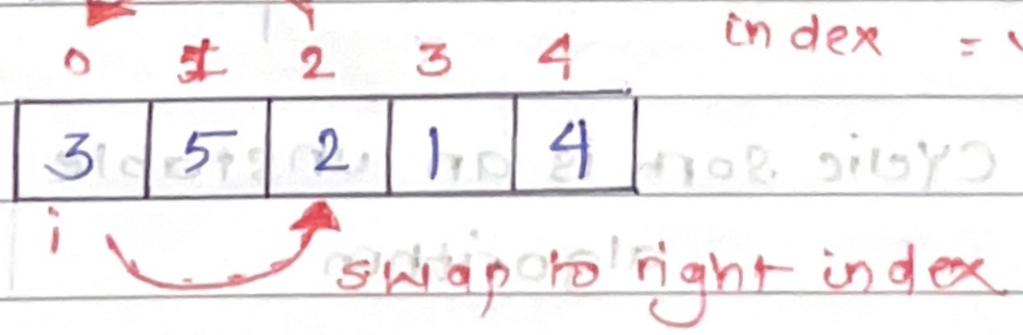
Complete: Keep going untill all numbers are there in corrected positions.

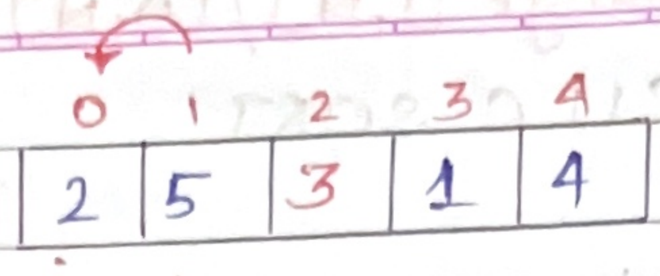
CHECK - SWAP - MOVE

* Index of value -

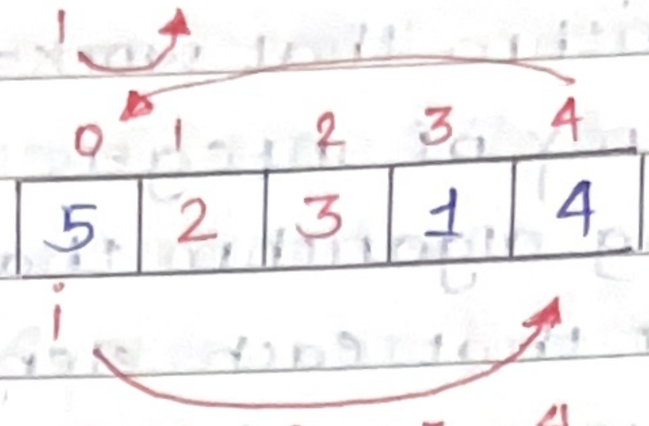
- If given number from Range [0,N] → index = value - 1
- If given number from Range [1,N] → index = value

[1, N]

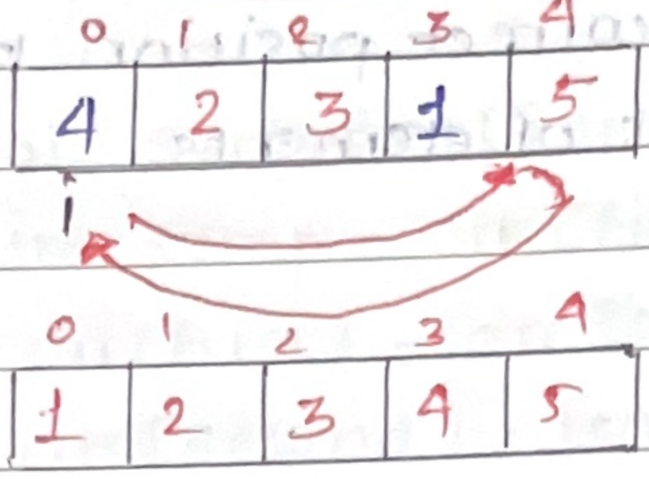




ith index = value - 1 = 2 - 1
= 1



ith index = value - 1 = 5 - 1
= 4



ith index = value - 1 = 4 - 1
= 3

Sorted Array

- Time Complexity -

Best Case / Worst Case ⟶ O(N)

Total comparisons: N - 1. (swap)
: (N - 1) + N
: (2N - 1)
O(N)

- We know that every element 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

Space complexity -

Auxiliary Space ⟶ O(1)

The space complexity is constant cause this algorithm is in place so it does not use any extra memory to sort.

Cyclic sort is an unstable algorithm

Example-

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {5, 4, 3, 2, 1};
        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 second){
        int temp = arr[first];
        arr[first] = arr[second];
        arr[second] = arr temp;
    }
}
```

O/P [1, 2, 3, 4, 5]