Cycle Sort

Given an

$$\text{Array} = \begin{bmatrix} 3 & 5 & 2 & 1 & 4 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

* When we know the range of the numbers in an Array, use Cyclic Sort

When range of numbers is (1, N) ⇒ use Cyclic Sort

Cyclic Sort Algorithm

$$\text{Initial Index} = \begin{bmatrix} 3 & 5 & 2 & 1 & 4 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

The range of numbers = (1, 5)

$$\text{After Sort} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

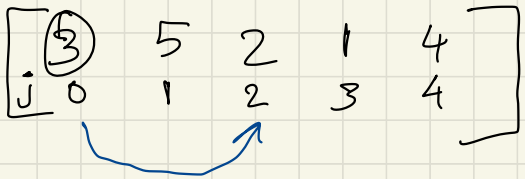
(i = 0)

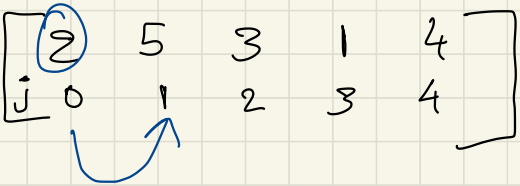
Value at index = Index + 1
(because index starts at zero)

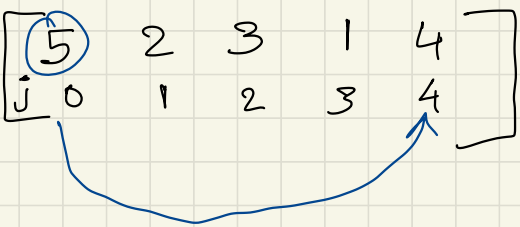
Value = Index + 1

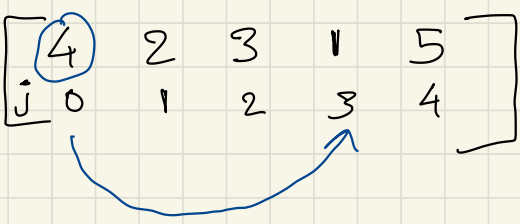
Index = Value − 1

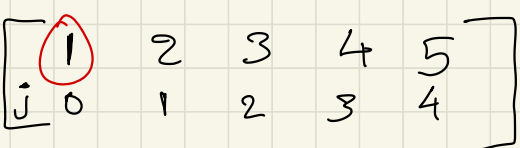
Start at $j = 0$



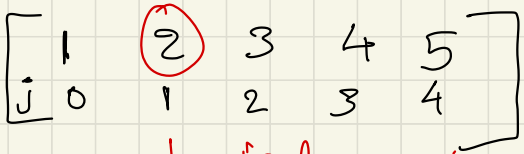








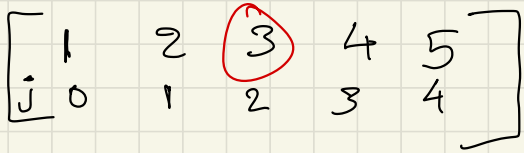
Now if the element at the first index is Correct move to the second Position and check $\Rightarrow$ $j = j + 1$



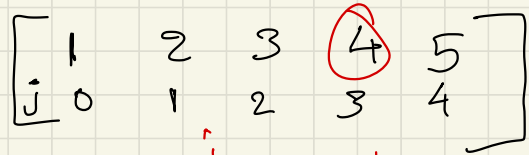
Value at index 2 is Correct

$\Rightarrow j = j+1$

$j = 1+1 = 2$



$j = 2+1 = 3$



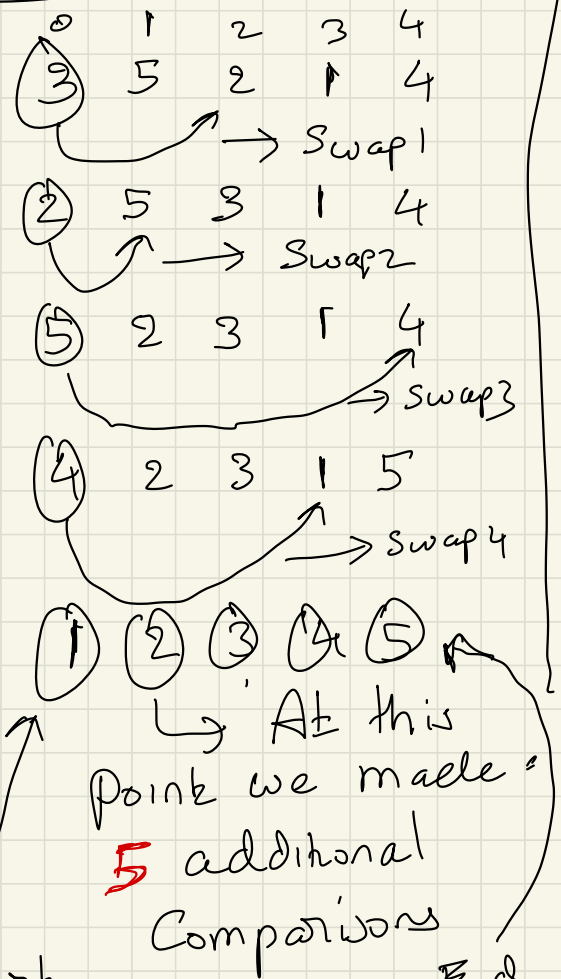
$j = 3+1 = 4$

Exit loop when $(j = N-1)$

→ As since all element till "N-2" index is already in the Correct place, we don't have to check the number at "N-1" index

Time Complexity

Worst Case example



→ Total 5 Comparisons.

Total
= 4 swaps made
+ 5 Comparisons
= 9

$= N - 1 + N$

$= (2N - 1)$

$= O(N)$