Cycle Sort

Lecture-33

Raghav Garg

Today's checklist

1) Speciality
2) Where to use
3) Algorithm
4) Questions

Speciality of Cyclic Sort

↓	T.C.
B.S, S.S, I.S →	$O(n^2)$
M.S, Q.S →	$O(n \log n)$
C.S →	$O(n)$ → but only for some selective problem

Where to use ?

↓

array → [1,n] , [0,n] , → 1 to n → pe kucch kaam karna hai

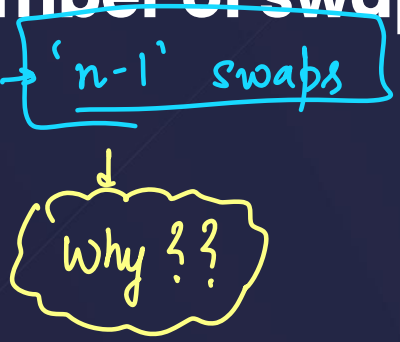
↳ T.C. O(n) & S.C O(1)

Algorithm for Cyclic Sort → It looks like it is useless

```
      0  1  2  3  4
a = { 5, 1, 2, 4, 3 }
→   { 3, 1, 2, 4, 5 }
→   { 2, 1, 3, 4, 5 }
→   { 1, 2, 3, 4, 5 }
```

```
      0  1  2  3  4  5
a = { 5, 1, 6, 3, 4, 2 }
→   { 4, 1, 6, 3, 5, 2 }
→   { 3, 1, 6, 4, 5, 2 }
→   { 6, 1, 3, 4, 5, 2 }
→   { 2, 1, 3, 4, 5, 6 }
→   { 1, 2, 3, 4, 5, 6 }
```

Ques : What is the worst number of swaps in Cyclic sort for an length n ?



Ques: Missing Number [Leetcode - 268]

M-I {Extra Space}

nums = {9, 6, 4, 2, 3, 5, 7, 0, 1}

Tne → n
tn+1

→ T.C. = O(n)

S.C = O(n)

	0	1	2	3	4	5	6	7	8	9
check =	1	1	1	1	1	1	1	1	0	1

Ques: Missing Number [Leetcode - 268]

M-2 : 'Cycle Sort'

	0	1	2	3	4	5	6	7	8
nums =	9	6	4	2	3	5	7	0	1
=	9	6	4	2	3	5	7	0	1
=	9	7	4	2	3	5	6	0	1
=	9	0	4	2	3	5	6	7	1
=	0	9	4	2	3	5	6	7	1
=	0	9	4	2	3	5	6	7	1
=	0	9	3	2	4	5	6	7	1
=	0	9	2	3	4	5	6	7	1

Ques: Missing Number [Leetcode - 268]

0 1 2 3 4 5 6 7 8

= {6, 9, 2, 3, 4, 5, 6, 7, 1}

= {0, 1, 2, 3, 4, 5, 6, 7, 9}

T.C. → O(n)

S.C. → O(1)

Ques: Find the duplicate number [Leetcode - 287]

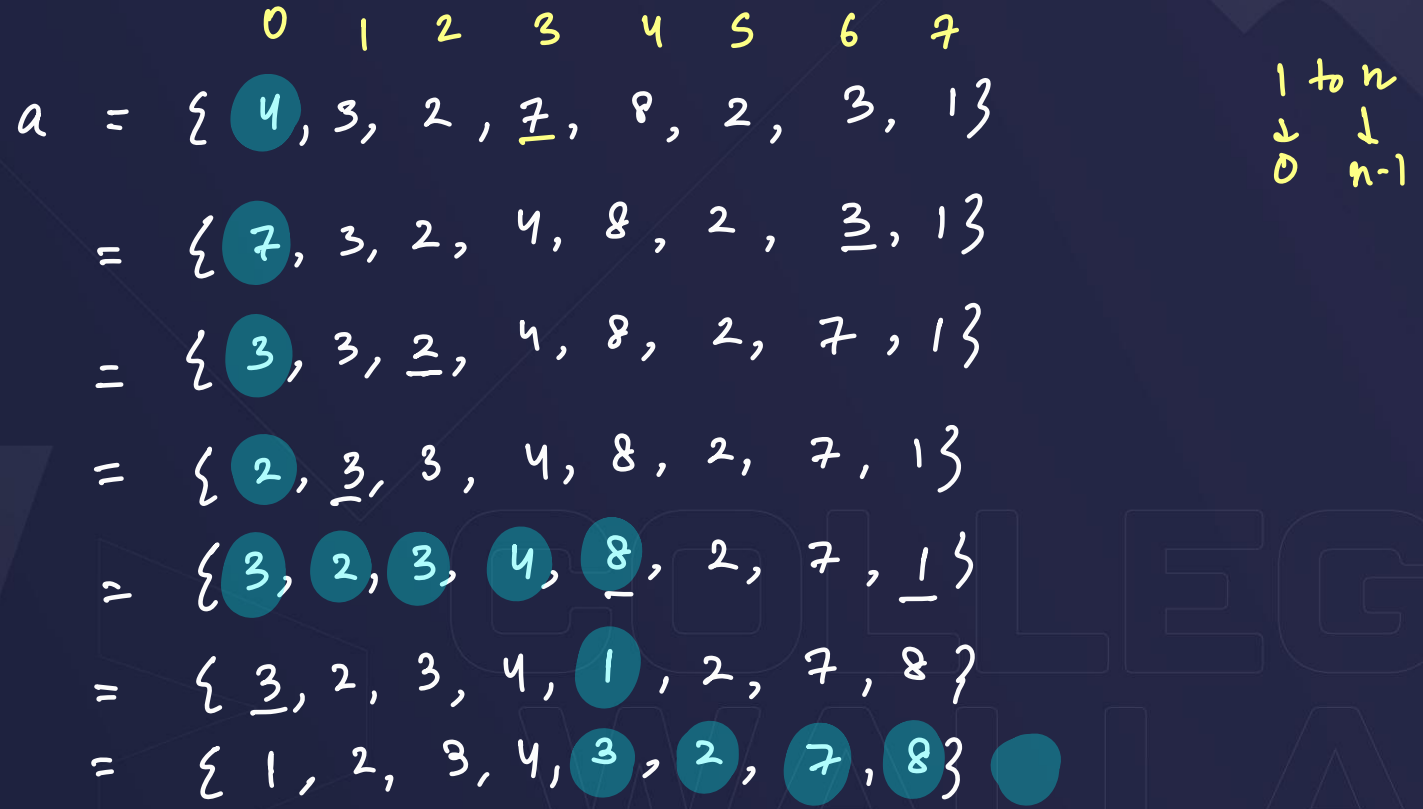
	0	1	2	3	4
a =	3	1	3	4	2
=	4	1	3	3	2
=	2	1	3	3	4
=	3	1	2	3	4

↓

duplicate

Ques: Find all numbers disappeared in an array

[Leetcode - 448]



Ques: First Missing Positive [Leetcode - 41]

'Very Famous' 'Interview'

→ T.C. = O(n)

S.C. = O(1)

correctIdx = nums[i] - 1;

+ve && <=n

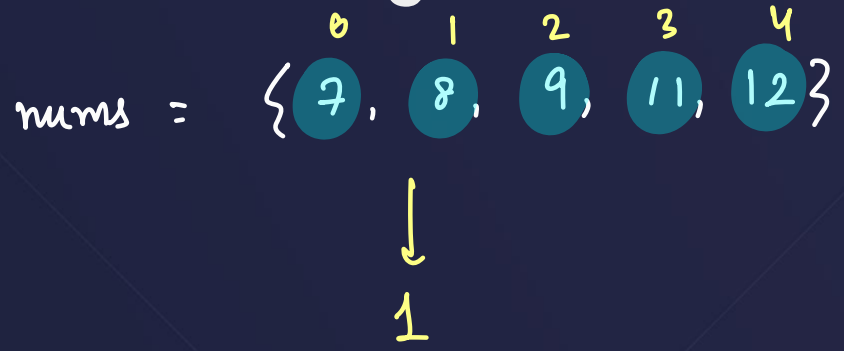
Ex-1

	0	1	2	3
nums =	3	4	-1	1
=	-1	4	3	1
=	-1	4	3	1
=	-1	1	3	4
=	1	-1	3	4
=	1	-1	3	4
=	1	-1	3	4

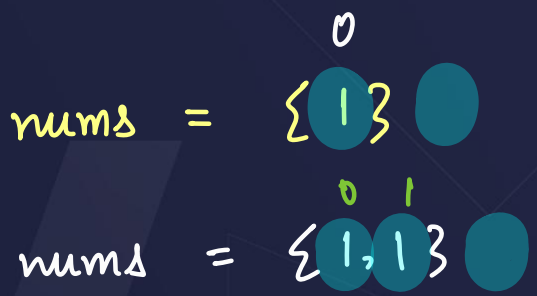
Ques: First Missing Positive

$n \rightarrow 5$

[Leetcode - 41]



cIdx = nums[i] - 1



Ques: First Missing Positive [Leetcode - 41]

```cpp
int firstMissingPositive(vector<int>& nums) {
    int n = nums.size(); 2
    int i = 0;
    // {1,1}
    while(i<n){
        int correctIdx = nums[i] - 1; // 0
        if(nums[i]<=0 || nums[i]>n || nums[correctIdx]==i+1) i++;
        else(swap(nums[i],nums[correctIdx]));
    }
    for(int i=0;i<n;i++){
        if(nums[i]!=i+1) return i+1;
    }
    return n+1;
}
```

0 1

nums = {1, 1}

i

cIdx = 0

THANK YOU