Grokking the Coding Interview: Patterns for Coding Questions

51% completed

Search Course

- Find the Duplicate Number (easy)
- Find all Duplicate Numbers (easy)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2
- Problem Challenge 3
- Solution Review: Problem Challenge 3

Pattern: In-place Reversal of a LinkedList

- Introduction
- Reverse a LinkedList (easy)
- Reverse a Sub-list (medium)
- Reverse every K-element Sub-list (medium)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2

Pattern: Tree Breadth First Search

- Introduction
- Binary Tree Level Order Traversal (easy)
- Reverse Level Order Traversal (easy)
- Zigzag Traversal (medium)
- Level Averages in a Binary Tree (easy)
- Minimum Depth of a Binary Tree (easy)
- Level Order Successor (easy)
- Connect Level Order Siblings (medium)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2

Pattern: Tree Depth First Search

- Introduction
- Binary Tree Path Sum (easy)
- All Paths for a Sum (medium)
- Sum of Path Numbers (medium)
- Path With Given Sequence (medium)
- Count Paths for a Sum (medium)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem

Explore

Tracks

Solution Review: Problem Challenge 2

We'll cover the following

- Find the Smallest Missing Positive Number (medium)
- Solution
- Code
  - Time complexity
  - Space complexity

Find the Smallest Missing Positive Number (medium)

Given an unsorted array containing numbers, find the **smallest missing positive number** in it.

Example 1:

```
Input: [-3, 1, 5, 4, 2]
Output: 3
Explanation: The smallest missing positive number is '3'
```

Example 2:

```
Input: [3, -2, 0, 1, 2]
Output: 4
```

Example 3:

```
Input: [3, 2, 5, 1]
Output: 4
```

Solution

This problem follows the **Cyclic Sort** pattern and shares similarities with Find the Missing Number with one big difference. In this problem, the numbers are not bound by any range so we can have any number in the input array.

However, we will follow a similar approach though as discussed in Find the Missing Number to place the numbers on their correct indices and ignore all numbers that are out of the range of the array (i.e., all negative numbers and all numbers greater than or equal to the length of the array). Once we are done with the cyclic sort we will iterate the array and the first index that does not have the correct number will be the smallest missing positive number!

Code

Here is what our algorithm will look like:

Java | Python3 | C++ | JS

```
function find_first_missing_positive(nums) {
  let i = 0;
  n = nums.length;
  while (i < n) {
    j = nums[i] - 1;
    if (nums[i] > 0 && nums[i] <= n && nums[i] !== nums[j]) {
      [nums[i], nums[j]] = [nums[j], nums[i]]; // swap
    } else {
      i += 1;
    }
  }
  for (i = 0; i < n; i++) {
    if (nums[i] !== i + 1) {
      return i + 1;
    }
  }

  return nums.length + 1;
}


console.log(find_first_missing_positive([-3, 1, 5, 4, 2]));
console.log(find_first_missing_positive([3, -2, 0, 1, 2]));
console.log(find_first_missing_positive([3, 2, 5, 1]));
```

RUN SAVE RESET

Close

Output 2.495s

```
3
4
4
```

Tracks

My Courses

Edpresso

Refer a Friend

Create

Challenge 2

Pattern: Two Heaps

● Introduction
● Find the Median of a Number Stream (medium)
● Sliding Window Median (hard)
○ Maximize Capital (hard)
○ Problem Challenge 1
○ Solution Review: Problem Challenge 1

Pattern: Subsets

○ Introduction
● Subsets (easy)
● Subsets With Duplicates (easy)
● Permutations (medium)
○ String Permutations by changing case (medium)
○ Balanced Parentheses (hard)
○ Unique Generalized Abbreviations (hard)
○ Problem Challenge 1
○ Solution Review: Problem

Time complexity

The time complexity of the above algorithm is $O(n)$.

Space complexity

The algorithm runs in constant space $O(1)$.

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. See how

Back
Problem Challenge 2

Next
Problem Challenge 3

☑ Mark as Completed

Report an Issue   Ask a Question