Documentation: First Missing Positive

Introduction

The "First Missing Positive" problem requires finding the smallest positive integer missing from an unsorted integer array. This documentation provides an overview of the problem, its constraints, and the solution approach to achieve the desired outcome. The solution is implemented in Python and adheres to the specified requirements of running in O(n) time complexity and using O(1) auxiliary space.

Problem Statement

Given an unsorted integer array `nums`, the task is to return the smallest positive integer that is not present in the array.

Example Scenarios:

Example 1:

Input: nums = [1, 2, 0]

Output: 3

Explanation: All numbers in the range [1, 2] are present in the array. The smallest positive integer missing is 3.

Example 2:

Input: nums = [3, 4, -1, 1]

Output: 2

Explanation: While 1 is present in the array, 2 is missing.

Example 3:

Input: nums = [7, 8, 9, 11, 12]

Output: 1

Explanation: The smallest positive integer missing is 1.

Constraints:

- The length of `nums` is between 1 and 10^5.
- Each element `nums[i]` is within the range [-2^31, 2^31 - 1].

Solution Approach

The solution employs an algorithm that runs in O(n) time complexity and utilizes O(1) auxiliary space. It follows two main steps:

1. **Cyclic Sort:** Iterate through the array to sort it in a cyclic manner. In each iteration, swap elements to ensure that each positive integer in the range [1, n] (where n is the length of the array) occupies its correct index (i.e., nums[i] = i + 1). This step ensures that all positive integers from 1 to n are placed at their respective positions.

2. **Finding Missing Integer:** After cyclic sorting, iterate through the array again. The first index where `nums[i] != i + 1` indicates the smallest positive integer missing from the array. If no such element is found, the missing integer is `n + 1`.

Implementation

The solution is implemented in Python using a class named `Solution`. The `firstMissingPositive` method takes an unsorted integer array `nums` as input and returns the smallest missing positive integer. The implementation strictly adheres to the specified time and space complexity requirements.

Test Cases:

- Test cases are provided to validate the correctness of the solution.
- These test cases cover various scenarios to ensure the solution handles different input cases accurately.

Conclusion

The "First Missing Positive" problem is effectively solved using a two-step approach that ensures optimal time and space complexity. By following the outlined solution, one can accurately determine the smallest positive integer missing from an unsorted integer array while meeting the given constraints.