Documentation for Longest Consecutive Sequence Problem

Description

Given an unsorted array of integers nums, return the length of the longest sequence of consecutive elements. The solution must have an algorithmic complexity of O(n).

Example 1

- **Input:** nums = [100, 4, 200, 1, 3, 2]
- **Output:** 4
- **Explanation:** The longest consecutive sequence is [1, 2, 3, 4], which has a length of 4.

Example 2

- **Input:** nums = [0, 3, 7, 2, 5, 8, 4, 6, 0, 1]
- **Output:** 9
- **Explanation:** The longest consecutive sequence is [0, 1, 2, 3, 4, 5, 6, 7, 8], which has a length of 9.

Constraints

- 0 <= nums.length <= 10^5
- -10^9 <= nums[i] <= 10^9

Overview

To solve the problem in O(n) time, we use a set to achieve O(1) average-time complexity for lookups. The algorithm involves the following steps:

1. **Early Exit:** If the input array nums is empty, return 0 immediately.
2. **Convert to Set:** Convert the input list nums to a set num_set to facilitate O(1) lookups.
3. **Initialize Longest Streak:** Initialize a variable longest_streak to keep track of the longest consecutive sequence found.
4. **Iterate Through Set:** Iterate through each number in num_set. For each number, check if it is the start of a sequence by verifying if the previous number (num 1) is not in num_set.
    - Count Consecutive Sequence: If the number is the start of a sequence, count the length of the consecutive sequence by incrementing the number and checking if the next number exists in the set.
    - Update Longest Streak: Update longest_streak if the current sequence is longer than the previously recorded longest sequence.
5. **Return Result:** Return the value of longest_streak.

Detailed Steps

1. Early Exit:

- If nums is empty, return 0.

2. Convert to Set:

- Convert nums to a set num_set to allow O(1) time complexity for element checks.

3. Initialize Longest Streak:

- Initialize longest_streak to 0.

4. **Iterate Through Set:**
   - *For each number num in num_set:*
     - Check if num 1 is not in num_set to identify the start of a sequence.
     - *If it is the start of a sequence:*
       - ✓ Initialize current_num to num and current_streak to 1.
       - ✓ While current_num + 1 is in num_set, increment current_num and current_streak.
       - ✓ Update longest_streak with the maximum value between longest_streak and current_streak.

5. **Return Result:**
   - Return longest_streak.

This approach ensures that each number is processed at most twice (once for the start of the sequence and once during sequence counting), resulting in O(n) time complexity.