[Click on the logo to find **Problem Statement**]

Intuition

The code aims to find the length of the longest consecutive sequence of numbers in an input list (`nums`). It does so by first creating a set (`numset`) to efficiently check for the presence of numbers. It then iterates through each number in the input list, and for each number, it checks if its predecessor is not in the set. If so, it initiates a while loop to count the consecutive numbers present in the set, updating the length of the current sequence. The code keeps track of the longest sequence encountered.

Approach

1. **Create a Set:**
    - Create a set (`numset`) to efficiently check for the presence of numbers in the input list. This set will be used to determine whether a number's predecessor is in the set.
2. **Iterate through Numbers:**
    - Iterate through each number (`n`) in the input list.
3. **Check for Predecessor:**
    - For each number, check if its predecessor (`n-1`) is not present in the set. This indicates the potential starting point of a consecutive sequence.
4. **Find Consecutive Sequence:**
    - If the predecessor is not in the set, initiate a while loop to count the consecutive numbers present in the set, starting from the current number (`n`). Increment a length variable until a gap is encountered in the consecutive sequence.
5. **Update Longest Sequence:**
    - Update the `longest` variable with the maximum length encountered during the iteration. This ensures that the code keeps track of the longest consecutive sequence seen so far.
6. **Return Result:**

- After iterating through all numbers, return the value of the `longest` variable, representing the length of the longest consecutive sequence.

Complexity

- Time complexity: $O(n^2)$
- Space complexity: O(n)

Code

```
class Solution:
    def longestConsecutive(self, nums: List[int]) -> int:
        numset = set(nums)
        longest = 0

        for n in nums:
            if (n-1) not in numset:
                length = 0
                while (n + length) in numset:
                    length += 1
                longest = max(length, longest)
        return longest
```

If you want to see more solutions to coding problems, you can visit: