

Largest Consecutive

For this problem we need an $O(n)$ solution, so sorting ($O(n \log n)$) is not allowed.

[Key Idea:]

Hash Set + Sequence Start Detection

1. Store all numbers in hash set for $O(1)$ lookup.
2. For each number num :
 - Only start counting if $num - 1$ is not in the set (meaning num is the start of sequence).
 - Count consecutive numbers $num + 1, num + 2, \dots$, until they are missing from the set.
3. Track the maximum length found.

[Algorithm:]

- Insert all numbers into `unordered_set<int>`.
- Iterate through numbers:
 - If $num - 1$ not in set:
 - Initialize `currentNum = num` and `currentLength = 1`.
 - While `currentNum + 1` is in set:
 - Increment both `currentNum` and `currentLength`.
 - Update `maxLength` with `currentLength`.

[Complexity:]

- Time: $O(n)$ - each number is processed at most twice (check start, extend sequence).
- Space: $O(n)$ - hash set stores all numbers.