Interview Questions and Answers

# Question 1:

- Given a list of integers, write a Python function to find the longest consecutive sequence. For example, if the input is `[100, 4, 200, 1, 3, 2]`, the output should be `[1, 2, 3, 4]`. Explain your approach and the time complexity of your solution.

## Answer:

To solve the problem of finding the longest consecutive sequence in a list of integers, you can follow these steps:  
  
1. \*\*Convert the list to a set\*\*: This allows for O(1) average time complexity for membership checks.  
2. \*\*Iterate through the list\*\*: For each number, check if it is the start of a sequence (i.e., the number minus one is not in the set).  
3. \*\*Find the length of the sequence\*\*: If the number is the start of a sequence, find the length of the sequence by checking for consecutive numbers.  
4. \*\*Update the longest sequence\*\*: Keep track of the longest sequence found.  
  
Here is the Python function implementing this approach:  
  
```python  
def longest\_consecutive(nums):  
 if not nums:  
 return []  
  
 num\_set = set(nums)  
 longest\_seq = []  
  
 for num in nums:  
 # Check if this number is the start of a sequence  
 if num - 1 not in num\_set:  
 current\_num = num  
 current\_seq = []  
  
 # Find the length of the sequence  
 while current\_num in num\_set:  
 current\_seq.append(current\_num)  
 current\_num += 1  
  
 # Update the longest sequence if the current one is longer  
 if len(current\_seq) > len(longest\_seq):  
 longest\_seq = current\_seq  
  
 return longest\_seq  
  
# Example usage  
nums = [100, 4, 200, 1, 3, 2]  
print(longest\_consecutive(nums)) # Output: [1, 2, 3, 4]  
```  
  
### Explanation of the Approach  
  
1. \*\*Convert the list to a set\*\*: This step ensures that membership checks (`num - 1 not in num\_set`) are efficient.  
2. \*\*Iterate through the list\*\*: For each number, check if it is the start of a sequence by verifying that `num - 1` is not in the set.  
3. \*\*Find the length of the sequence\*\*: If the number is the start of a sequence, keep incrementing the number and adding it to the current sequence until the number is not found in the set.  
4. \*\*Update the longest sequence\*\*: If the current sequence is longer than the previously found longest sequence, update the longest sequence.  
  
### Time Complexity  
  
- \*\*Converting the list to a set\*\*: O(n), where n is the number of elements in the list.  
- \*\*Iterating through the list and finding sequences\*\*: Each number is checked at most twice (once when it is the start of a sequence and once when it is part of a sequence), so the overall time complexity is O(n).  
  
Therefore, the time complexity of the solution is O(n).

--------------------------------------------------

# Question 2:

- In your "Chest Disease Classification from Chest CT Scan Image" project, you mentioned using ResNet50 architecture. Can you explain why ResNet50 was a suitable choice for this task? Additionally, how did you handle the class imbalance issue, if any, in your dataset?

## Answer:

The provided context does not include details on why ResNet50 was chosen or how class imbalance was handled in the "Chest Disease Classification from Chest CT Scan Image" project. Therefore, I don't know the answers to these questions.

--------------------------------------------------