Time Complexity

This accurately explains the run time analysis (Worst-Case Big-O Notation) for each solution we produced.

Task 0

Texts

The time complexity of accessing a specific index in a list is constant, O(1). Therefore, accessing the first element of the 'texts' list has constant time complexity.

Calls

The main task is retrieving the last element of the 'calls' list. This can be done by finding the list's length and accessing the element at the calculated index. Since both of these operations have constant time complexities, the overall time complexity of this procedure is O(1).

The provided solution for both tasks has a constant runtime complexity of O(1) because the operations involve simple index access in lists, and the input size does not affect the execution time.

Runtime Complexity: O(1)

Task 1

Iterating Through Texts and Calls

The time complexity of the loops is proportional to the number of texts (n) and calls (m).

Adding to Distinct List

Within the loop, there are two checks - one for the caller/sender and another for the receiver - for each record.
If there are n text and m call records, adding to the distinct list has a complexity of O(n+m).

Final Message

Printing the final message has a constant time complexity.

The main factor is iterating through the text and call records (n + m).

Runtime Complexity: O(n + m)

Task 2

Building the dictionary phone_time_spent

The operations performed on each call record, such as dictionary updates or insertions, take a constant amount of time.
The complexity of building a dictionary increases linearly with the number of call records. If there are 'n' call records, the time complexity for building the dictionary will be O(n).

Finding the Longest Time and Corresponding Phone Number

Iterating through a dictionary to find the phone number with the longest time has a linear time complexity since it involves going through all entries in the dictionary once.
If there are n entries in the dictionary, the time complexity to find the longest entry is O(n).

Iterating through call records (n) is the main factor of complexity.

Runtime Complexity: O(n)

Task 3

Iterating Through Calls

The time complexity is proportional to the number of calls (n) because the loop iterates through all call records.

Checking and Extracting Codes

The code identifies the call type (Fixed line, Mobile, Telemarketer) and extracts the corresponding code.
The operations for performing checks and extraction are constant time operations, which involve string slicing or finding operations that are also constant time.
The complexity of this part is O(1) for each call, meaning that it has a constant time complexity regardless of the input size.

List Operations:

Each call to append to the 'area_code' list and check for duplicates has constant time complexity.
If there is a list of m unique codes, the time complexity for performing any operation on this list is O(m).

Sorting

Sorting the 'area_code' list has a time complexity of O(m log m). This means that the time required to sort the list grows logarithmically with the number of elements in the list. In other words, as the number of elements in the list increases, the time required to sort the list increases much slower.
The sorting operation is secondary when iterating through call records (n).

Runtime Complexity: O(n + m log m)

Task 4

Iterating Through Calls

The loop iterates through all call records, so the time complexity is proportional to the number of calls (n).

Iterating Through Texts

The loop iterates through all text records, so the time complexity is proportional to the number of texts (m).

Set Operations

The set operations (addition, subtraction) have constant time complexity for each call or text record.
In the worst-case scenario, the overall complexity for the set operations is O(n + m).

Sorting:

Sorting the 'suspects' list has a time complexity of O(t log t), where t is the number of possible telemarketers.

Print:

The last part of the code involves printing the list of suspected telemarketers (suspects) by iterating through each element of the list and printing it. This operation has a linear time complexity since each element of the list is iterated only once. As t is the number of suspected telemarketers, the complexity of printing suspects is O(t).

Overall, the dominant factor is the iteration through the call records (n) and text records (m), and the set operations and sorting are secondary.
With O(2n + 2m + t log t + t), we ended up with O(n + m + t log t) after simplification.

Runtime Complexity: O(n + m + t log t)