Task 0 Time Complexity: O(1)
Explanation:
Returning and reading an element at an index only requires constant time complexity. It doesn't involve iteration for the indexes of the elements in this case are already known.

Task 1 Time Complexity: O(n)
Explanation:
In order to return the length of a list of unique telephone numbers, the four columns, telephone numbers to and from, are the data. To return all the unique elements requires iterating through the entire data, and removing the duplicates, or returning a dictionary's length with the indexes as the telephone numbers. Either implementation would vary the time complexity if the data's length varies as each new element would need to be appended.

Task 2 Time Complexity: O(n)
Explanation:
In order to return the user who was on for the longest period would involve iterating until the end, unable to guarantee whether there is a future value greater than the present one. The time complexity is that of the data's length ; it would need to include all the values.

Task 3 Time Complexity: O(nlog(n))
Explanation:
To return the percentage of telephone users from Bangalore to Bangalore out of Bangalore would require iteration throughout the entire data. If a new element is appended to the data, the value could increment the new value and alter the return value given the previous percentage, or re-iterate through all the data. To return the data in lexicographic order as required would require a sorting algorithm, which would have a nlogn time complexity.

Task 4 Time Complexity: O(nlog(n))
Explanation:
A telemarketer would have to satisfy the following conditions, not to receive messages, not to be either a caller or receiver in the text message data. After iterating through each number and appending the numbers that satisfy the requirements, return the numbers. To return the data in

lexicographic order as required would require a sorting algorithm, which would have a nlogn time complexity.