Task0:

|  |
| --- |
| Time Complexity: O(1) |
|  |

|  |
| --- |
| Description : This exercise only requires access to a specific value in the array. So, no matter the size of the input, it only executes one line. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Task1: |
|  |

|  |
| --- |
| Time Complexity: O(n) + O(k) |
|  |

|  |
| --- |
| Description : This solution uses a set structure from a list. set() is essentially iterating over the list and adding each of those elements to a hash table. In this exercise, the time is proportional to input size for calls (n) and messages (k). |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Task2: |
|  |

|  |
| --- |
| Time Complexity: O(n) |
|  |

|  |
| --- |
| Description : The solution was to iterate each phone and store them in a list to make a relation with the total time spent. In the same iteration a compare is made to get the longest time. One iteration was required so this solution is proportional to the input size. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Task3: |
|  |

|  |
| --- |
| Time Complexity: O(nlogn) |
|  |

|  |
| --- |
| Description : The solution compares all items from list in a single iteration, so is proportional to the size of the input size (O(n)). But a sort function is called and the time complexity for this one is O(nlog(n)). The approximation results with O(n) and O(nlog(n)), but the higher time complexity is taken, so in this case is O(nlog(n)). |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Task4: |
|  |

|  |
| --- |
| Time Complexity: O(nlogn) |
|  |

Description: Same as before, the solution compares all items in a single iteration. But a sort function is called and time complexity results with O(nlog(n)).