Unscramble Computer Science Problems

REVIEW

CODE REVIEW

HISTORY

Requires Changes

3 specifications require changes

Dear Learner,

Awesome work is done in this submission. You have completed the tasks amazingly well.
Overall brilliant work.

However, there are some modifications needed, kindly follow the comments in the corresponding specifications.

I believe you will have further fun and a learning experience working through them.
Feel free to post your specific doubts at "https://knowledge.udacity.com/"

Keep learning.
Good luck !!!

Rubric

Your code should be well-structured and readable.

Great, your code is neat and properly structured.

Recommendation

Avoid variables like i, j, etc. Kindly give more meaningful names.

Print only the solution outputs. Feel free to use other print statements during the development process, but remember to remove them for submission.

The output format is absolutely correct.

Task 0 - The script correctly prints out the information of first record of texts and last record of calls.

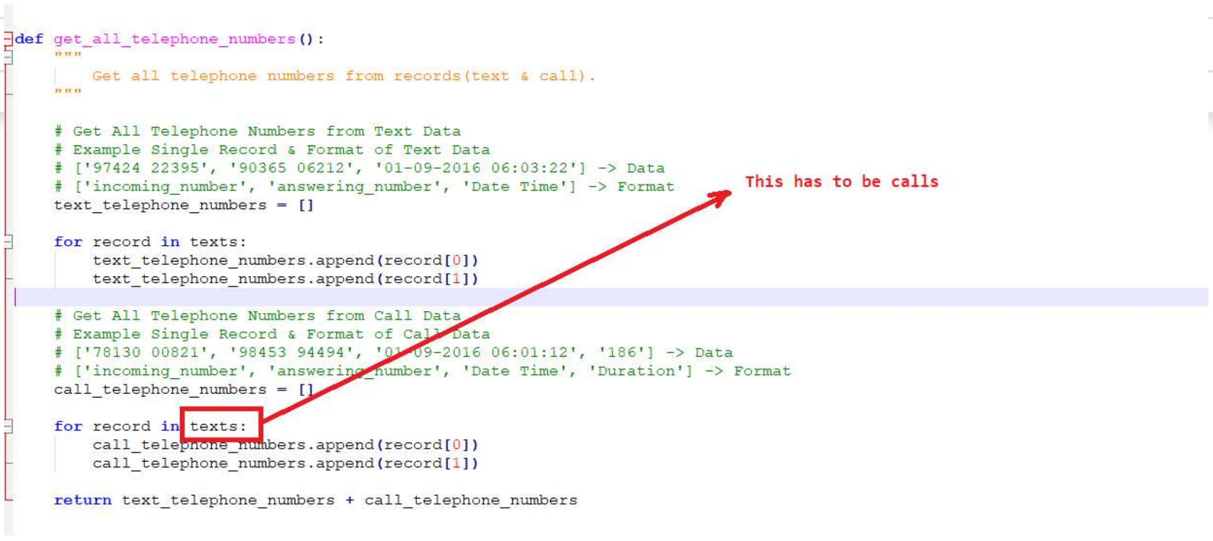
The script correctly prints out the information of first record of texts and last record of calls. ✅

Task 1 - The script correctly prints number of distinct telephone numbers in the dataset.

The answer for this task is incorrect.

Hint

The error came due to a typo. Kindly correct that:-



```
def get_all_telephone_numbers():
    """
        Get all telephone numbers from records(text & call).
    """

    # Get All Telephone Numbers from Text Data
    # Example Single Record & Format of Text Data
    # ['97424 22395', '90365 06212', '01-09-2016 06:03:22'] -> Data
    # ['incoming_number', 'answering_number', 'Date Time'] -> Format
    text_telephone_numbers = []

    for record in texts:
        text_telephone_numbers.append(record[0])
        text_telephone_numbers.append(record[1])

    # Get All Telephone Numbers from Call Data
    # Example Single Record & Format of Call Data
    # ['78130 00821', '98453 94494', '01-09-2016 06:01:12', '186'] -> Data
    # ['incoming_number', 'answering_number', 'Date Time', 'Duration'] -> Format
    call_telephone_numbers = []

    for record in texts:
        call_telephone_numbers.append(record[0])
        call_telephone_numbers.append(record[1])

    return text_telephone_numbers + call_telephone_numbers
```

Task 2 - The script correctly prints the telephone number that spent the longest time on the phone and the total time in seconds they spend on phone call.

Task 3 - The script correctly prints the telephone codes called by fixed-line numbers in Bangalore and the percentage of calls from fixed lines in Bangalore that are to fixed lines in Bangalore.

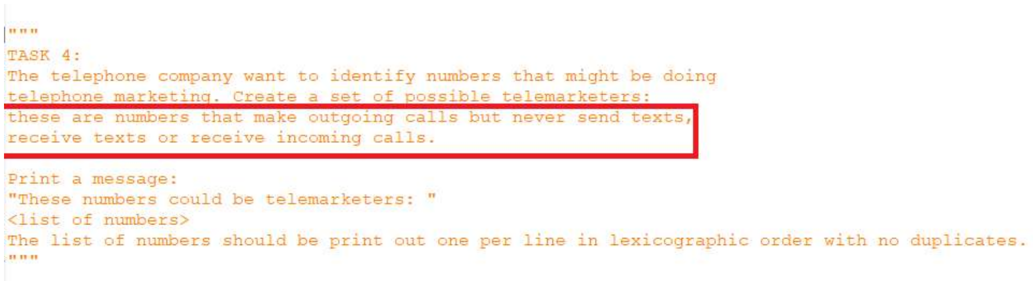
Awesome, both the Parts i.e PART-A and PART-B runs absolutely fine

Task 4 - The script correctly prints the list of numbers that could be telemarketers.

The answer for this task is incorrect

Hint

Kindly follow the instructions given in this task to complete the task:-

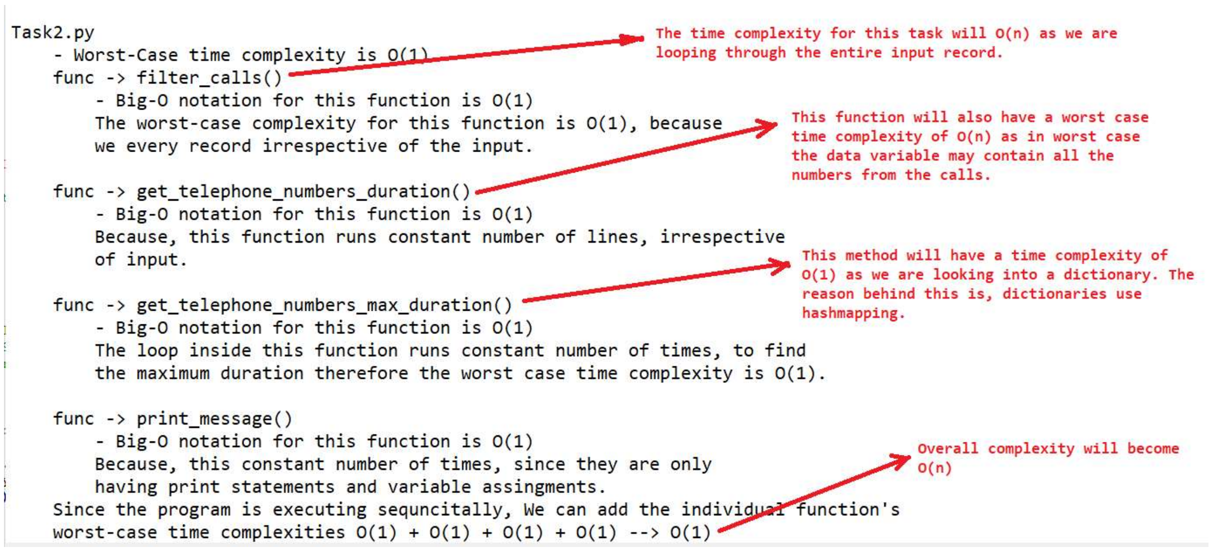


```
"""
TASK 4:
The telephone company want to identify numbers that might be doing
telephone marketing. Create a set of possible telemarketers:
these are numbers that make outgoing calls but never send texts,
receive texts or receive incoming calls.

Print a message:
"These numbers could be telemarketers: "
<list of numbers>
The list of numbers should be print out one per line in lexicographic order with no duplicates.
"""
```

Student provides a text file accurately explaining their run time analysis (Worst-Case Big-O Notation) for each solution they produced.

Task 2:-



```
Task2.py
    - Worst-Case time complexity is O(1)
    func -> filter_calls()
        - Big-O notation for this function is O(1)
        The worst-case complexity for this function is O(1), because
        we every record irrespective of the input.

    func -> get_telephone_numbers_duration()
        - Big-O notation for this function is O(1)
        Because, this function runs constant number of lines, irrespective
        of input.

    func -> get_telephone_numbers_max_duration()
        - Big-O notation for this function is O(1)
        The loop inside this function runs constant number of times, to find
        the maximum duration therefore the worst case time complexity is O(1).

    func -> print_message()
        - Big-O notation for this function is O(1)
        Because, this constant number of times, since they are only
        having print statements and variable assingments.
    Since the program is executing sequncitally, We can add the individual function's
    worst-case time complexities O(1) + O(1) + O(1) + O(1) --> O(1)
```

The time complexity for this task will O(n) as we are looping through the entire input record.

This function will also have a worst case time complexity of O(n) as in worst case the data variable may contain all the numbers from the calls.

This method will have a time complexity of O(1) as we are looking into a dictionary. The reason behind this is, dictionaries use hashmapping.

Overall complexity will become O(n)

Task3:-

For this task, you have used the sorted method, the time complexity for which will be O(nlogn).

You need to consider this as well.

Resources

To have better grasp on time complexity, kindly follow:-

- https://towardsdatascience.com/understanding-time-complexity-with-python-examples-2bda6e8158a7
- Screencast1
- Screencast2
- https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.