**Worst Case Analysis Of Tasks**

**Task-0: Printing out the information of first record of texts and last record of calls**.

**Explaination:**

* **Statements:** Statements like “print”, were used in the program. The time complexity of statements is constant time i,e O(1)

**Time Complexity: O(1)**

**Space Complexity: O(1)**

**Task-1: Printing the number of distinct telephone numbers in the dataset.**

**Explaination:**

* **Statements:** Statements like “print”, were used in the program. The time complexity of statements is constant time i,e O(1)
* **Code Snippet:**

*for record in records:*

*rec\_set.add(record[0])*

*rec\_set.add(record[1])*

1. **Loops:** “For loop” was used in the program. Complexity is O(n)
2. **In-built Python Function:** set is used, using the “add” method. Complexity: O(1) i,e inside For-Loop : O(n\*1)=O(n)

**Time Complexity: O(n)**

**Space Complexity: O(n)**

**Task-2: Printing the telephone number that spent the longest time on the phone and the total time in seconds they spend on phone call.**

**Explaination:**

* **Statements:** Statements like “print”, were used in the program. The time complexity of statements is constant time i,e O(1)
* **Code Snippet:**

*for record in calls:*

*pn1 = record[0]*

*pn2 = record[1]*

1. **Loops:** “For loop” was used in the program. Complexity is O(n)
2. **If-Else statements inside for loop:**
3. *if pn1 in my\_dict.keys():*

*my\_dict[pn1] += call\_duration*

*else:*

*my\_dict[pn1] = call\_duration*

1. *if pn2\_d >= pn1\_d:*

*if max\_duration < pn2\_d:*

*max\_duration = pn2\_d*

*max\_d\_pn = pn2*

*else:*

*if max\_duration < pn1\_d:*

*max\_duration = pn1\_d*

*max\_d\_pn = pn1*

* a) and b) worst case time complexity is O(1) I,e total time-complexity O(n\*constant)=O(n)

**Time Complexity: O(n)**

**Space Complexity:O(n)**

**Task-3: Printing 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.**

**Explaination:**

* **Statements:** Statements like “print”, were used in the program. The time complexity of statements is constant time i,e O(1)
* **CodeSnippet:**

*for record in calls:*

1. *if record[0][:5] == "(080)":*

*total\_count += 1*

*pn = record[1]*

*b) if(pn[0] == "("):*

*if pn[:5] == "(080)":*

*banglore\_count += 1*

*area\_code = pn.split(")")[0]*

*area\_codes\_list.add(area\_code[1:])*

*c) elif (pn[0] in ['7','8','9']):*

*area\_code = pn[:4]*

*area\_codes\_list.add(area\_code)*

*d)  else:*

*area\_codes\_list.add("140")*

1. **Loops:** “For loop” was used in the program. Complexity is O(n)
2. **If-Else statements inside for loop: The** statements a,b,c,d inside if loop take constant time I,e O(1)
3. **In-built function (Split function):** Time complexity: O(1)
4. **In-built function (set.add):** Time complexity: O(1)

Inside For-loop: O(n)

* **In-built function (sorted)**: Time complexity: O(nlogn)

O(n)<O(nlogn)

**Time Complexity: O(n^logn)**

**Space Complexity:O(n)**

**Task-4: Printing the list of numbers that could be telemarketers.**

**Explaination:**

* **Statements:** Statements like “print”, were used in the program. The time complexity of statements is constant time i,e O(1)
* **Code Snippet:**

*for record in calls:*

*pn = record[0]*

*if pn not in my\_set:*

*telemarketing\_list.add(pn)*

1. **Loops:** “For loop” was used in the program. Complexity is O(n)
2. **If sentence:** Time complexity: O(1)
3. **In-built function(set.add):** Time Complexity: O(1)

For-loop Complexity: O(n\*constant)=O(n)

* **In-built function (sorted)**: Time complexity: O(nlogn)

O(n)<O(n^logn)

**Time Complexity: O(n^logn)**

**Space Complexity: O(n)**