

Name: Ahaan Desai
Roll no.: 231070016
Batch: Batch A (1-20)

DAA Lab Experiment 3

Aim: To find the maximum and minimum gross and net salary of employees using linear algorithm as well as divide and conquer technique.

Program:

```
Python
import numpy as np
import pandas as pd
from typing import List

class User:
    def __init__(self, user_id, basic_salary):
        self.user_id = user_id
        self.basic_salary = basic_salary
        self.hra = int(0.4 * basic_salary)           # House Rent Allowance
        - 40% of basic salary
        self.income_tax = (
            0 if self.basic_salary <= 10000 else
            int(0.1 * self.basic_salary) if self.basic_salary <= 20000 else
            int(0.2 * self.basic_salary) if self.basic_salary <= 30000 else
            int(0.3 * self.basic_salary)
        )
        self.provident_fund = int(0.12 * self.basic_salary)

    def calculate_gross_salary(self):
        return self.basic_salary + self.hra

    def calculate_net_salary(self):
        return self.calculate_gross_salary() - self.income_tax -
self.provident_fund

class CalculateMaximumAndMinimum():
```

```

def __init__(self, users_array: List[User]):
    self.users = users_array
    self.size = len(self.users)
    self.gross_salaries = [user.calculate_gross_salary() for user in
self.users]
    self.net_salaries = [user.calculate_net_salary() for user in
self.users]
    self.__show_result(self.gross_salaries, type="gross")
    self.__show_result(self.net_salaries, type="net")

def __find_max_min_iterative(self, array):
    """Finds maximum and minimum salary out of an array of 2000
employees."""
    max_salary = array[0]
    min_salary = array[0]
    for i in range(1, len(array)):
        max_salary = max(max_salary, array[i])
        min_salary = min(min_salary, array[i])

    return max_salary, min_salary

def __find_max_divconq(self, array, start, end):
    """Finds maximum salary out of an array of 2000 employees using divide
and conquer approach."""
    if start > end:
        return -1
    elif start == end:
        return array[start]
    elif end - start + 1 == 2:
        return max(array[start], array[end])

    mid = (start + end) // 2
    return max(
        self.__find_max_divconq(array, start, mid-1),
        self.__find_max_divconq(array, mid+1, end),
        array[mid]
    )

def __find_min_divconq(self, array, start, end):
    """Finds minimum salary out of an array of 2000 employees using divide
and conquer approach."""
    if start > end:
        return float('inf')
    elif start == end:

```

```

        return array[start]
    elif end - start + 1 == 2:
        return min(array[start], array[end])

    mid = (start + end) // 2
    return min(
        self.__find_min_divconq(array, start, mid-1),
        self.__find_min_divconq(array, mid+1, end),
        array[mid]
    )

def __find_user_id(self, value, array):
    index = np.argmax(array == value)
    return self.users[index].user_id

def __show_result(self, array, type):
    max_salary_iterative, min_salary_iterative =
self.__find_max_min_iterative(array)
    max_salary_divconq = self.__find_max_divconq(array, 0, self.size - 1)
    min_salary_divconq = self.__find_min_divconq(array, 0, self.size - 1)

    assert max_salary_iterative == max_salary_divconq
    assert min_salary_iterative == min_salary_divconq

    employee_with_maximum_salary =
self.__find_user_id(max_salary_iterative, array)
    employee_with_minimum_salary =
self.__find_user_id(min_salary_iterative, array)

    print(f"Maximum {type} salary by iterative method:
{max_salary_iterative}")
    print(f"Maximum {type} salary found by divide and conquer:
{max_salary_divconq}")
    print(f"Employee ID with maximum salary:
{employee_with_maximum_salary}")
    print(f"Minimum {type} salary by iterative method:
{min_salary_iterative}")
    print(f"Minimum {type} salary found by divide and conquer:
{min_salary_divconq}")
    print(f"Employee ID with minimum salary: {employee_with_minimum_salary}
\n")

class ProcessData():
    def __init__(self, filename):

```

```

self.filename = filename
self.data = None
self.size = None
self.users = None

print(f"Results for file {self.filename}")
self.__read_csv()
if self.__check_data() == -1: return
self.__build_users()
CalculateMaximumAndMinimum(self.users)

def __read_csv(self):
    try:
        self.data = pd.read_csv(self.filename).dropna(subset=['User ID',
'Basic Salary']).to_numpy()
        self.size = len(self.data)
        self.users = [None] * self.size
    except pd.errors.EmptyDataError:
        print("File is empty!")
        return

def __build_users(self):
    self.users = [User(self.data[i][0], self.data[i][1]) for i in
range(self.size)]

def __check_data(self):
    if self.data is None:
        print("Data is not read!")
        return

    user_ids = self.data[:, 0]          # first column of dataframe
    basic_salaries = self.data[:, 1]    # second column of dataframe

    if len(user_ids) != 2000 or len(basic_salaries) != 2000:
        print("Column sizes are invalid! \n")
        return -1

    if np.any(basic_salaries <= 0) or np.any(user_ids <= 0):
        print("Invalid data! \n")
        return -1

    return 1

```

```
if __name__ == '__main__':  
    ProcessData("datasetA.csv")  
    ProcessData("datasetB.csv")  
    ProcessData("datasetC.csv")  
    ProcessData("datasetD.csv")  
    ProcessData("datasetE.csv")
```

Output:

```
ahaandesai@DESKTOP-UI6FGCD:~/Labs/DAA/Salary Calculation New$ python3
Results for file datasetA.csv
Maximum gross salary by iterative method: 69991
Maximum gross salary found by divide and conquer: 69991
Employee ID with maximum salary: 996
Minimum gross salary by iterative method: 14050
Minimum gross salary found by divide and conquer: 14050
Employee ID with minimum salary: 926

Maximum net salary by iterative method: 48994
Maximum net salary found by divide and conquer: 48994
Employee ID with maximum salary: 996
Minimum net salary by iterative method: 11843
Minimum net salary found by divide and conquer: 11843
Employee ID with minimum salary: 926

Results for file datasetB.csv
Maximum gross salary by iterative method: 69972
Maximum gross salary found by divide and conquer: 69972
Employee ID with maximum salary: 1252
Minimum gross salary by iterative method: 14081
Minimum gross salary found by divide and conquer: 14081
Employee ID with minimum salary: 330

Maximum net salary by iterative method: 48981
Maximum net salary found by divide and conquer: 48981
Employee ID with maximum salary: 1252
Minimum net salary by iterative method: 11870
Minimum net salary found by divide and conquer: 11870
Employee ID with minimum salary: 330

Results for file datasetC.csv
Maximum gross salary by iterative method: 69986
Maximum gross salary found by divide and conquer: 69986
Employee ID with maximum salary: 460
Minimum gross salary by iterative method: 14037
Minimum gross salary found by divide and conquer: 14037
Employee ID with minimum salary: 1370

Maximum net salary by iterative method: 48991
Maximum net salary found by divide and conquer: 48991
Employee ID with maximum salary: 460
Minimum net salary by iterative method: 11832
Minimum net salary found by divide and conquer: 11832
Employee ID with minimum salary: 1370

Results for file datasetD.csv
Column sizes are invalid!

Results for file datasetE.csv
Invalid data!
```

Conclusion:

We have implemented the divide and conquer technique to find the maximum and minimum gross and net salary of employees. The divide and conquer technique is a fundamental algorithmic strategy used to solve complex problems by breaking them into smaller, solvable subproblems. These subproblems are then merged together to form the solution to the original problem.