Practical No 7

Aim: CPU Scheduling Algorithm (Part 1)- FCFS and Non-Preemptive Scheduling

```
def fcfsScheduling(processes):
    """
    processes: list of tuples (pid, arrivalTime, burstTime)
    """
    # Sort by arrival time
    processes.sort(key=lambda x: x[1])

    startTime = []
    completionTime = []
    waitingTime = []
    turnaroundTime = []
    currentTime = 0
    ganttChart = []

    for pid, arrival, burst in processes:
        if currentTime < arrival:
            currentTime = arrival  # CPU idle until process arrives

        startTime.append(currentTime)
        ganttChart.append((pid, currentTime, currentTime + burst))
        currentTime += burst
        completionTime.append(currentTime)

        tat = completionTime[-1] - arrival
        wt = tat - burst

        turnaroundTime.append(tat)
        waitingTime.append(wt)

    avgWt = sum(waitingTime) / len(processes)
    avgTat = sum(turnaroundTime) / len(processes)

    # Print results
    print("\n--- FCFS Scheduling ---")
    print("PID\tAT\tBT\tST\tCT\tTAT\tWT")
    for i, p in enumerate(processes):
        print(f"{p[0]}\t{p[1]}\t{p[2]}\t{startTime[i]}\t{completionTime[i]}\t{turnaroundTime[i]}\
t{waitingTime[i]}")

    print(f"\nAverage Waiting Time: {avgWt:.2f}")
    print(f"Average Turnaround Time: {avgTat:.2f}")

    print("\nGantt Chart:")
    for pid, start, end in ganttChart:
        print(f"| P{pid} ({start}-{end}) ", end="")
    print("|")
```

```
# Example usage
processes = [
    (1, 0, 5),
    (2, 2, 3),
    (3, 4, 1)
]

fcfsScheduling(processes)
```

```
--- FCFS Scheduling ---
PID    AT     BT     ST     CT     TAT    WT
1      0      5      0      5      5      0
2      2      3      5      8      6      3
3      4      1      8      9      5      4

Average Waiting Time: 2.33
Average Turnaround Time: 5.33

Gantt Chart:
| P1 (0-5) | P2 (5-8) | P3 (8-9) |
```

7.2 Extend implementation to general non-preemptive scheduling.

7.3 Analyze waiting time, turnaround time, and Gantt chart generation.

```
def nonPreemptivePriority(processes):
    """
    processes: list of tuples (pid, arrivalTime, burstTime, priority)
    Lower priority value means higher priority.
    """
    n = len(processes)

    # Sort by arrival time first, then priority
    processes.sort(key=lambda x: (x[1], x[3]))

    completed = 0
    currentTime = 0
    startTime = {}
    completionTime = {}
    waitingTime = {}
    turnaroundTime = {}
    ganttChart = []
    readyQueue = []
    visited = [False] * n

    while completed < n:
        # Add processes that have arrived by current time
        for i in range(n):
            if processes[i][1] <= currentTime and not visited[i]:
                readyQueue.append(processes[i])
                visited[i] = True

        if readyQueue:
            # Pick highest priority (lowest number = highest priority)
            readyQueue.sort(key=lambda x: x[3])
            pid, at, bt, pr = readyQueue.pop(0)
            if currentTime < at:
                # CPU idle until process arrives
                currentTime = at

            startTime[pid] = currentTime
            ganttChart.append((pid, currentTime, currentTime + bt))
            currentTime += bt
            completionTime[pid] = currentTime

            turnaroundTime[pid] = completionTime[pid] - at
            waitingTime[pid] = turnaroundTime[pid] - bt
            completed += 1
        else:
            # No process ready → CPU idle
            currentTime += 1

    avgWt = sum(waitingTime.values()) / n
    avgTat = sum(turnaroundTime.values()) / n

    # Print Results
    print("\n--- Non-Preemptive Priority Scheduling ---")
    print("PID\tAT\tBT\tPriority\tST\tCT\tTAT\tWT")
    for pid, at, bt, pr in processes:
        print(f"{pid}\t{at}\t{bt}\t{pr}\t\t{startTime[pid]}\t{completionTime[pid]}\t{turnaroundTime[pid]}\
t{waitingTime[pid]}")

    print(f"\nAverage Waiting Time: {avgWt:.2f}")
    print(f"Average Turnaround Time: {avgTat:.2f}")

    print("\nGantt Chart:")
    for pid, start, end in ganttChart:
        print(f"| P{pid} ({start}-{end}) ", end="")
    print("|")

# Example usage
processesPriority = [
    (1, 0, 5, 2),  # (PID, Arrival Time, Burst Time, Priority)
    (2, 1, 3, 1),
    (3, 2, 8, 3),
    (4, 3, 6, 2)
]

nonPreemptivePriority(processesPriority)
```

```
--- Non-Preemptive Priority Scheduling ---
PID   AT    BT    Priority    ST    CT    TAT   WT
1     0     5     2           0     5     5     0
2     1     3     1           5     8     7     4
3     2     8     3           14    22    20    12
4     3     6     2           8     14    11    5

Average Waiting Time: 5.25
Average Turnaround Time: 10.75

Gantt Chart:
| P1 (0-5) | P2 (5-8) | P4 (8-14) | P3 (14-22) |
```