# SJF (Shortest Job First) Scheduling Algorithm - Non-preemptive

processes = [

{'pid': 'P1', 'arrival': 0, 'burst': 6},

{'pid': 'P2', 'arrival': 1, 'burst': 8},

{'pid': 'P3', 'arrival': 2, 'burst': 7},

{'pid': 'P4', 'arrival': 3, 'burst': 3},

]

n = len(processes)

completed = 0

current\_time = 0

waiting\_times = [0] \* n

turnaround\_times = [0] \* n

is\_completed = [False] \* n

process\_order = []

print("PID\tArrival\tBurst\tStart\tFinish\tWaiting\tTurnaround")

while completed < n:

idx = -1

min\_burst = float('inf')

for i in range(n):

if (processes[i]['arrival'] <= current\_time) and (not is\_completed[i]):

if processes[i]['burst'] < min\_burst:

min\_burst = processes[i]['burst']

idx = i

elif processes[i]['burst'] == min\_burst:

if processes[i]['arrival'] < processes[idx]['arrival']:

idx = i

if idx != -1:

start\_time = current\_time

finish\_time = start\_time + processes[idx]['burst']

waiting\_time = start\_time - processes[idx]['arrival']

turnaround\_time = finish\_time - processes[idx]['arrival']

waiting\_times[idx] = waiting\_time

turnaround\_times[idx] = turnaround\_time

current\_time = finish\_time

is\_completed[idx] = True

completed += 1

process\_order.append((processes[idx]['pid'], start\_time, finish\_time))

print(f"{processes[idx]['pid']}\t{processes[idx]['arrival']}\t{processes[idx]['burst']}\t{start\_time}\t{finish\_time}\t{waiting\_time}\t{turnaround\_time}")

else:

current\_time += 1 # If no process is available, increment time

avg\_waiting = sum(waiting\_times) / n

avg\_turnaround = sum(turnaround\_times) / n

print(f"\nAverage Waiting Time: {avg\_waiting:.2f}")

print(f"Average Turnaround Time: {avg\_turnaround:.2f}")

# Gantt Chart

print("\nGantt Chart:")

for pid, start, finish in process\_order:

print(f"| {pid} ({start}-{finish}) ", end="")

print("|")