B. Case Study 2: Greedy Algorithms - Job Scheduling

6. Implement a Python program to schedule jobs and minimize the total completion time.

```python
def schedule_jobs(jobs):
    # Sort the jobs by their duration (processing time)
    sorted_jobs = sorted(jobs)

    total_completion_time = 0
    current_time = 0

    # Schedule the jobs in the sorted order
    for job in sorted_jobs:
        current_time += job
        total_completion_time += current_time

    return total_completion_time

# Example list of job durations
jobs = [3, 1, 4, 2]

# Compute the minimum total completion time
min_completion_time = schedule_jobs(jobs)
print("Minimum Total Completion Time:", min_completion_time)
```