Problem Statement

Project Title:
Smart Daily & Study Planner: An Algorithmic Approach to Optimizing Student Productivity and Study Time

Background and Motivation:
University students face daily challenges in managing their time efficiently. A typical student day includes multiple competing activities such as attending lectures, completing assignments, studying for exams, personal tasks, and taking necessary breaks. Poor time allocation often leads to stress, burnout, and reduced academic performance.
Most students rely on manual planning or simple to-do lists, which fail to consider important constraints such as limited time, mental fatigue, task priorities, and varying subject difficulty. Therefore, there is a need for a smart, automated system that can generate an optimal daily schedule and study plan based on algorithmic principles.
This project proposes a Smart Daily & Study Planner that models time management as an optimization problem and applies different algorithm design paradigms taught in the course to solve it.

Problem Definition:
Given a set of daily tasks and study requirements, the goal is to generate an optimal schedule that:

- Maximizes overall productivity and expected academic benefit
- Respects time and fatigue constraints
- Ensures mandatory breaks are included

The problem combines daily task scheduling and study time optimization into a single unified framework.

Inputs:
The system receives the following inputs:
Daily Tasks:
For each task *i*:

- duration_i: Required time to complete the task (in hours)
- priority_i: Importance level of the task
- fatigue_i: Mental or physical effort required

Study Subjects:
For each subject *j*:

- study_time_j: Recommended study time
- difficulty_j: Difficulty level of the subject
- weight_j: Contribution of the subject to final grades

Global Constraints:

- T: Total available hours in the day
- F_max: Maximum allowable fatigue level
- Mandatory break durations

Outputs:
The system produces:

- A detailed daily schedule (hour-by-hour)
- Allocated study time for each subject
- Total productivity score
- Total fatigue score

Objective Function:
The objective is to maximize the total productivity score:
Maximize $\Sigma(\text{Productivity}_i) + \Sigma(\text{StudyBenefit}_j)$
Subject to:
$\Sigma(\text{Time}_i + \text{StudyTime}_j) \leq T$
$\Sigma(\text{Fatigue}_i + \text{StudyFatigue}_j) \leq F_{max}$

Algorithmic Approaches:
The problem is solved using multiple algorithmic paradigms:
Brute Force Approach:

- Tries all possible task and study orderings
- Guarantees optimal solution
- Computationally infeasible for large inputs

Greedy Approach:

- Selects tasks or subjects based on highest priority or difficulty first
- Fast but does not guarantee optimality

Dynamic Programming Approach:

- Models the problem as a constrained optimization problem
- Efficiently computes the optimal solution
- Balances productivity, time, and fatigue

Divide and Conquer Approach:

- Divides the day into smaller time segments
- Solves each segment independently
- Merges partial schedules into a full-day plan

Example:
Input Example

- Available time: 10 hours
- Tasks: Lecture (2h, priority 5), Assignment (3h, priority 8)
- Subjects: Algorithms (difficulty 9), Databases (difficulty 6)

Output Example

- Morning: Lecture + Algorithms study
- Afternoon: Assignment + Break
- Evening: Databases study

Significance of the Problem:
This problem is realistic, relevant to students' daily lives, and complex enough to demonstrate the strengths and weaknesses of different algorithmic paradigms. It provides a clear comparison between naive and optimized solutions while maintaining strong practical value.

Conclusion:
The Smart Daily & Study Planner serves as an effective case study for applying algorithm analysis and design techniques to a real-world problem. By comparing brute force, greedy, dynamic programming, and divide-and-conquer approaches, the project highlights the importance of choosing the right algorithmic strategy for complex optimization problems.