I will discuss my experience in developing a genetic algorithm program for scheduling. The program was developed to optimize a schedule for a small team of employees, considering their availability, professions, and preferences. The program data is stored in dictionaries and lists, and the genetic algorithm was implemented using Python.

**Challenges in Writing the Program**

The main challenge in developing the program was programming an appropriate fitness function that accurately reflected the constraints and preferences of the employees. I spent most of the time researching to ensure that the algorithm would produce optimal schedules following our given requirements. Another challenge was the size of the generated population, which was affected significantly as the number of employees and tasks were being randomized and mutated overtime.

**Development**

Despite the challenges, I am pleased to say that the development process was relatively smooth. The problem is mostly carrying out the correct structure for the genetic algorithms to work and function as expected. Making decision on generating 1 parent or 2 parents, and how the selection happens when mutating, are the ones that I hesitated on. Additionally, the Python programming language is intuitive and easy to learn, which made development more efficient.

**Evaluation of the Schedule**

After running the genetic algorithm, the program produced a schedule that met all the constraints and preferences of the employees. However, there were a few instances where the schedule did not seem optimal, and there were some tasks that were scheduled at odd times. This was likely due to the fitness function not accurately reflecting the preferences of the employees. Additionally, the genetic algorithm may have been trapped in a local minimum, preventing it from finding the global minimum sometimes, which I changed the way mutation works and it further increased the efficiency of the algorithms.

**Improvement of the Program and Fitness Function**

To improve the program, I would add more constraints and preferences to the fitness function to make it more accurate. Additionally, I would experiment with different genetic operators and parameters to see if they improve the quality of the schedule. For example, we can add distance from each class lecture from its room which allows us to also manage the time for staff to coordinate on campus during shifts. I would also consider implementing a multi-objective optimization approach to optimize for multiple objectives simultaneously. Finally, I would explore the use of other optimization techniques such as using an external library in Python or similar framework for genetic algorithm in scheduling.

**Conclusion**

In conclusion, developing a genetic algorithm program for scheduling was a challenging but rewarding experience. Despite encountering some difficulties, the program was successfully implemented and produced a feasible schedule.