Genetic Algorithm Report

1. Challenges
   1. For me, some of the challenges of this program were architecting, time, and algorithm logic. I started first before I started programming anything by laying out how the algorithm worked, how I wanted to implement it, what data structures to use, etc. I feel like this helped me in the long term as I didn’t have to refactor a lot of my code since most of it was thought out to begin with. However, it still took a significant amount of time and effort to fully dissect the problem and think of what the best data structure or method of processing would be best and most efficient for this problem. Additionally, it took me quite a while to understand how the CDF for crossover works and to implement it as well. So overall, this program took a good chunk of time for me to complete that I hadn’t originally estimated, but overall I’m happy with the results that it produces.
2. Output
   1. As for the program output, I feel as though it is pretty self explanatory with what classes, times, and professors are recommended for the ideal schedule. The thing I find interesting is how the fitness function and it’s constraints are shown in the output. For example, it appears to really prioritize having professors teach in consecutive timeslots, but no more than 3-4 in a row. Additionally, there appears to be no room location where a professor goes from bloch to katz or vise versa. Most of it is either in the quad or to the quad to bloch, quad to katz, etc.
3. Areas of improvement
   1. The only things I would change about my program are efficiency and organization. I feel like I started out with these goals in mind, but as I was implementing the program I just found something that worked and went with it rather than spending a lot of time trying to find the most efficient way of doing so. Additionally towards the end (especially in the crossover function) I feel that my code got a little messy and could be cleaned up a little bit, but at the same time I feel as though I’ve spent enough time on it and don’t want to spend even more time refactoring and potentially breaking something that works.
4. Final thoughts
   1. With that being said though, I am proud of the organization and structure of my program overall. Especially how I have different components for classes, have a fitness function class, and enums for professors, class times, and buildings. I feel like it made the program pretty modular and easy to work with throughout. Overall, even though this project took a lot of time, I really enjoyed it and getting to implement something not only of this scale but of this complexity.