# Intro

So basically, there were two search algorithms that needed to be implemented. One was the steepest ascent hill climb with random restart and other one was the genetic algorithm. These were needed to be implemented in generic way and not should be problem specific.

# Steepest Ascent hill climb With random restart

There were already helper functions written for each problem in test cases. I first initialized the population and then after that I ran a loop until my budget comes to end. Each time I get best neighbor from our population and then it is checked whether the new best neighbor is better than previous best neighbor, if it is then best value is updated with this neighbor.

# Genetic Algorithm

First we initialize the population and then we get two best parents from our population. After that I get new offsprings from these two parents by mutation or crossover. In mutation the one value in the replaced with the some random value from population. In crossover, two new offsprings are created by cutting parents and joining with each other. These processes run until budget is over.

# Review Assignment Problem

The review assignment problem was solved with genetic algorithm. First, I randomly initialized two lists with topics and students and their preference for each topic was also randomly initialized. Two covert data from phenotype to genotype I created a random list of topics equal to the problem size. After that for the process of mutation I just replaced a topic with a random topic in the list. After that for crossover the two individuals were cut from a random point and were replaced with each other.