AI Lab 07
Section D

Genetic algorithm developed by Goldberg was inspired by Darwin's theory of evolution which states that the survival of an organism is affected by rule "the strongest species that survives". Darwin also stated that the survival of an organism can be maintained through the process of reproduction, crossover and mutation. Darwin's concept of evolution is then adapted to computational algorithm to find solution to a problem called objective function in natural fashion. A solution generated by genetic algorithm is called a chromosome, while collection of chromosome is referred as a population. A chromosome is composed from genes and its value can be either numerical, binary, symbols or characters depending on the problem want to be solved. These chromosomes will undergo a process called fitness function to measure the suitability of solution generated by GA with problem. Some chromosomes in population will mate through process called crossover thus producing new chromosomes named offspring which its genes composition are the combination of their parent. In a generation, a few chromosomes will also mutation in their gene. The number of chromosomes which will undergo crossover and mutation is controlled by crossover rate and mutation rate value. Chromosome in the population that will maintain for the next generation will be selected based on Darwinian evolution rule, the chromosome which has higher fitness value will have greater probability of being selected again in the next generation. After several generations, the chromosome value will converges to a certain value which is the best solution for the problem.

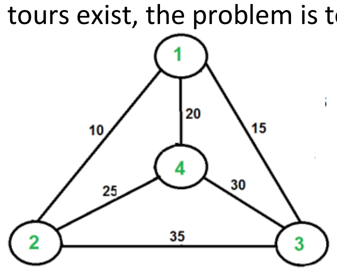
The genetic algorithm works as follows:

1. Determine the number of chromosomes, generation, and mutation rate and crossover rate value
2. Generate chromosome-chromosome number of the population, and the initialization value of the genes chromosome-chromosome with a random value
3. Process steps 4-7 until the number of generations is met
4. Evaluation of fitness value of chromosomes by calculating objective function
5. Chromosomes selection
6. Crossover Step
7. Mutation Step
8. Solution (Best Chromosomes)

You can read more about GA from here:
https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm

Your task is to solve the following problem by using Genetic algorithm:

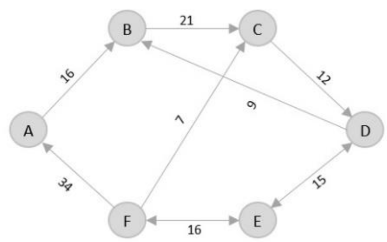
Given a set of cities and the distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point. Note the difference between Hamiltonian Cycle and this problem. The Hamiltonian cycle problem is to find if there exists a tour that visits every city exactly once. Here we know that Hamiltonian Tour exists (because the graph is complete) and in fact, many such tours exist, the problem is to find a minimum weight Hamiltonian Cycle.



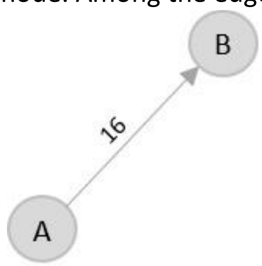
For example, consider the following graph. A TSP tour in the graph is 1-2-4-3-1. The cost of the tour is 10+25+30+15 which is 80. The problem is a famous NP-hard problem. There is no polynomial-time know solution for this problem.

Step by step Example by Greedy Approach:

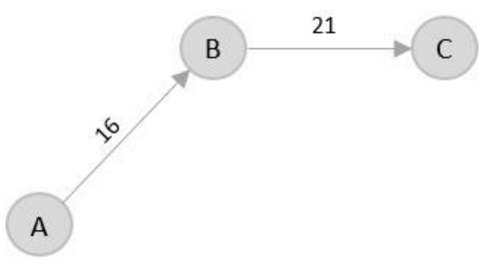
Consider the following graph with six cities and the distances between them −



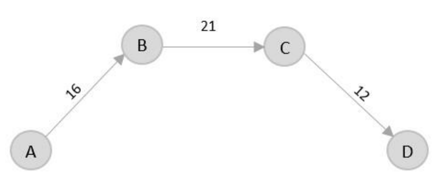
From the given graph, since the origin is already mentioned, the solution must always start from that node. Among the edges leading from A, A → B has the shortest distance.



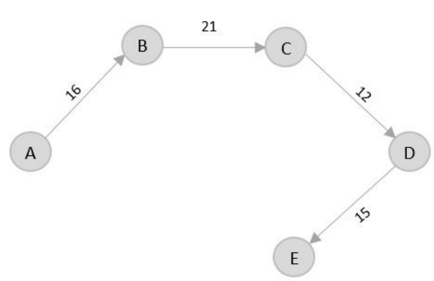
Then, B → C has the shortest and only edge between, therefore it is included in the output graph.



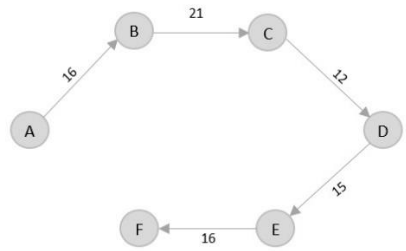
There's only one edge between C → D, therefore it is added to the output graph.



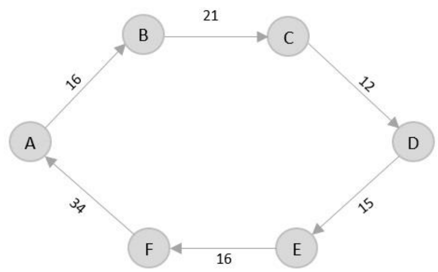
There's two outward edges from D. Even though, D → B has lower distance than D → E, B is already visited once and it would form a cycle if added to the output graph. Therefore, D → E is added into the output graph.



There’s only one edge from e, that is E → F. Therefore, it is added into the output graph.



Again, even though F → C has lower distance than F → A, F → A is added into the output graph in order to avoid the cycle that would form and C is already visited once.



The shortest path that originates and ends at A is A → B → C → D → E → F → A
The cost of the path is: 16 + 21 + 12 + 15 + 16 + 34 = 114.

The above step by step implementation is by Greedy approach, you have to implement this problem by using Genetic Algorithm.