README

Our project goal is to solve the Travelling Salesman Problem using genetic algorithm. In this project, four classes have been created, which are City class, Individual class, Population class and TSP class. The City class is used for initializing the location of each city representing by their coordinate. The Individual class contains the method of calculating and updating. It also including gene expression which it is named as route, standing for the tour between cities. The Population class is the most essential in this program, it has all the other aspects of GA, which are population, selection, crossover and mutation. It also has a sort function for sorting the best route, an evolution function and a logging function. Lastly, the TSP class is the main class, it involves all the GA parameters holds all the elements, and prints out the consequences.

Final Report

The Travelling Salesman Problem is mainly about calculating the shortest path among a list of cities. Therefore, we want firstly to have a list of 20 cities and calculate the distance among them. After the implementation of genetic algorithm, we can finally get the best route.

Class City

1. The City method initializes the coordinator for each city.

Class Individual

1. The getRouteWeight method can be considered as fitness, which calculate the total distance for each route by adding up the result getting from getTwoCitiesDistance method.
2. The getTwoCitiesDistance method calculate the distance of two random cities.

Class Population

1. The Evolve method produce the next generation by sorting and selecting the best DNA and taking into account of the possibility of mutation happens.
2. The Crossover method refers to combine father and mother’s DNA to produce the next generation. A portion of 30% will be selected from parents and then the portion from the

father will be added to the portion of mother's, which comes to the first child. By exchanging the rate, the second child will be created.

3. The Mutate method can be happened under a possibility of 10%, it will exchange one DNA with another.
4. The PopulationCutoff method sort the selected route, the shortest distance of the route will be considered as the best fitness.
5. The SelectCandidate method cut off 50% of the number of population and select the rest depends on their fitness.

Finding & Results:

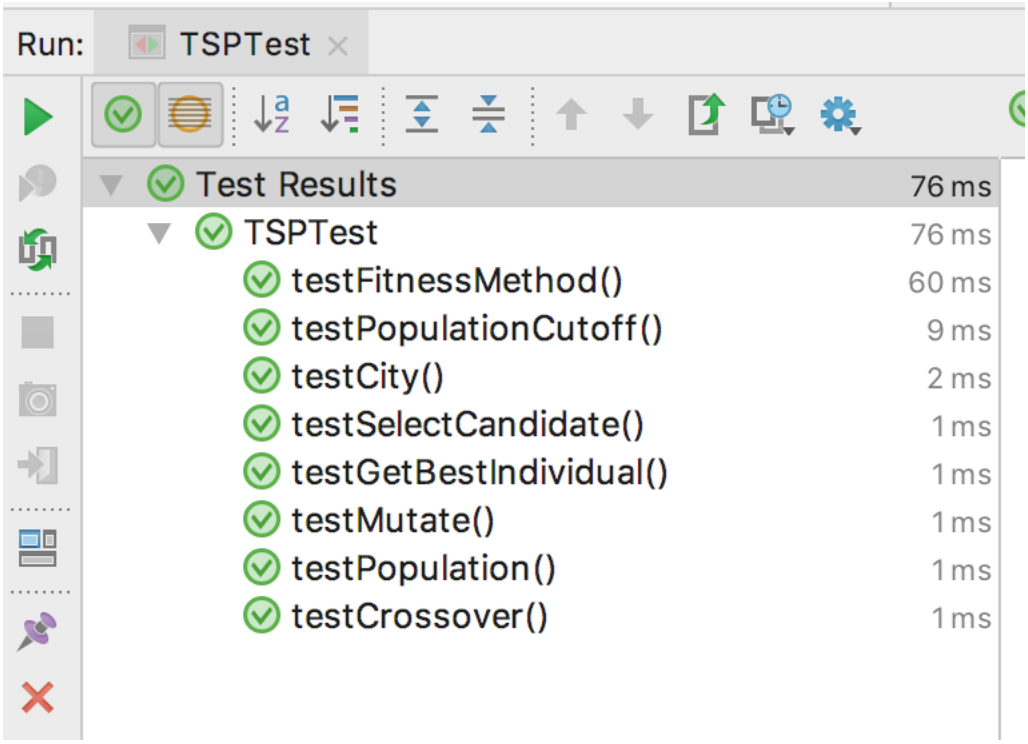
We calculate the best route among 20 customized cities, and the optimal result we got so far is 1936. However, the results are not stay the same for each running of the program, which is reasonable since we implement genetic algorithm. According to the logs, in the most cases, the resulting staring with gradually decrease and will remain unchanged at some point.

Conclusion:

The objective for our project is to find the shortest route among cities. Based on the unit test, all the methods of our program work properly. However, by using the genetic algorithm, it is difficult to write a test for verifying if the result is optimal. We believe we finally get the optimal one after many times of runs.

```
Generation:1 Fitness of the Best candidate: 4288
Generation:2 Fitness of the Best candidate: 4288
Generation:3 Fitness of the Best candidate: 4091
Generation:4 Fitness of the Best candidate: 3994
Generation:5 Fitness of the Best candidate: 3916
Generation:6 Fitness of the Best candidate: 3842
Generation:7 Fitness of the Best candidate: 3749
Generation:8 Fitness of the Best candidate: 3601
Generation:9 Fitness of the Best candidate: 3567
Generation:10 Fitness of the Best candidate: 3567
Generation:11 Fitness of the Best candidate: 3567
Generation:12 Fitness of the Best candidate: 3567
Generation:13 Fitness of the Best candidate: 3468
Generation:14 Fitness of the Best candidate: 3468
Generation:15 Fitness of the Best candidate: 3249
Generation:16 Fitness of the Best candidate: 3078
Generation:17 Fitness of the Best candidate: 2931
Generation:18 Fitness of the Best candidate: 2931
Generation:19 Fitness of the Best candidate: 2931
Generation:20 Fitness of the Best candidate: 2931
Generation:21 Fitness of the Best candidate: 2922
Generation:22 Fitness of the Best candidate: 2922
Generation:23 Fitness of the Best candidate: 2873
Generation:24 Fitness of the Best candidate: 2873
Generation:25 Fitness of the Best candidate: 2873
Generation:26 Fitness of the Best candidate: 2581
Generation:27 Fitness of the Best candidate: 2581
Generation:28 Fitness of the Best candidate: 2581
Generation:29 Fitness of the Best candidate: 2581
Generation:30 Fitness of the Best candidate: 2581
Generation:31 Fitness of the Best candidate: 2581
Generation:32 Fitness of the Best candidate: 2581
Generation:33 Fitness of the Best candidate: 2581
Generation:34 Fitness of the Best candidate: 2575
Generation:35 Fitness of the Best candidate: 2506
Generation:36 Fitness of the Best candidate: 2506
Generation:37 Fitness of the Best candidate: 2381
Generation:38 Fitness of the Best candidate: 2381
Generation:39 Fitness of the Best candidate: 2322
Generation:40 Fitness of the Best candidate: 2289
Generation:41 Fitness of the Best candidate: 2289
Generation:42 Fitness of the Best candidate: 2289
Generation:43 Fitness of the Best candidate: 2289
Generation:44 Fitness of the Best candidate: 2289
Generation:45 Fitness of the Best candidate: 2289
Generation:46 Fitness of the Best candidate: 2083
Generation:47 Fitness of the Best candidate: 2081
Generation:48 Fitness of the Best candidate: 1980
Generation:49 Fitness of the Best candidate: 1980
Generation:50 Fitness of the Best candidate: 1980
```

```
Generation:51 Fitness of the Best candidate: 1980
Generation:52 Fitness of the Best candidate: 1980
Generation:53 Fitness of the Best candidate: 1980
Generation:54 Fitness of the Best candidate: 1936
Generation:55 Fitness of the Best candidate: 1936
Generation:56 Fitness of the Best candidate: 1936
Generation:57 Fitness of the Best candidate: 1936
Generation:58 Fitness of the Best candidate: 1936
Generation:59 Fitness of the Best candidate: 1936
Generation:60 Fitness of the Best candidate: 1936
Generation:61 Fitness of the Best candidate: 1936
Generation:62 Fitness of the Best candidate: 1936
Generation:63 Fitness of the Best candidate: 1936
Generation:64 Fitness of the Best candidate: 1936
Generation:65 Fitness of the Best candidate: 1936
Generation:66 Fitness of the Best candidate: 1936
Generation:67 Fitness of the Best candidate: 1936
Generation:68 Fitness of the Best candidate: 1936
Generation:69 Fitness of the Best candidate: 1936
Generation:70 Fitness of the Best candidate: 1936
Generation:71 Fitness of the Best candidate: 1936
Generation:72 Fitness of the Best candidate: 1936
Generation:73 Fitness of the Best candidate: 1936
Generation:74 Fitness of the Best candidate: 1936
Generation:75 Fitness of the Best candidate: 1936
Generation:76 Fitness of the Best candidate: 1936
Generation:77 Fitness of the Best candidate: 1936
Generation:78 Fitness of the Best candidate: 1936
Generation:79 Fitness of the Best candidate: 1936
Generation:80 Fitness of the Best candidate: 1936
Generation:81 Fitness of the Best candidate: 1936
Generation:82 Fitness of the Best candidate: 1936
Generation:83 Fitness of the Best candidate: 1936
Generation:84 Fitness of the Best candidate: 1936
Generation:85 Fitness of the Best candidate: 1936
Generation:86 Fitness of the Best candidate: 1936
Generation:87 Fitness of the Best candidate: 1936
Generation:88 Fitness of the Best candidate: 1936
Generation:89 Fitness of the Best candidate: 1936
Generation:90 Fitness of the Best candidate: 1936
Generation:91 Fitness of the Best candidate: 1936
Generation:92 Fitness of the Best candidate: 1936
Generation:93 Fitness of the Best candidate: 1936
Generation:94 Fitness of the Best candidate: 1936
Generation:95 Fitness of the Best candidate: 1936
Generation:96 Fitness of the Best candidate: 1936
Generation:97 Fitness of the Best candidate: 1936
Generation:98 Fitness of the Best candidate: 1936
Generation:99 Fitness of the Best candidate: 1936
Generation:100 Fitness of the Best candidate: 1936
BUILD SUCCESSFUL (total time: 0 seconds)
```


Run:
TSPTest
Test Results 76 ms
TSPTest 76 ms
testFitnessMethod() 60 ms
testPopulationCutoff() 9 ms
testCity() 2 ms
testSelectCandidate() 1 ms
testGetBestIndividual() 1 ms
testMutate() 1 ms
testPopulation() 1 ms
testCrossover() 1 ms