INFO 6205: Program Structures & Algorithms

Genetic Algorithm: Simplified Vehicle Routing Problem

***Team Members:*** **Ivan Fernando, Suyog Sathe**

***Team Number:* 319**

**Problem:**

We found a solution to a Vehicle Routing Problem variant using Genetic Algorithm. The problem statement says, given a set of patients and distance between them and the source(hospital), find the minimum distance for a path that delivers patients to the hospital. This can also be stated as given a complete graph, find shortest path using Dijkstra's Algorithms.

**Solution/Implementation:**

**Fitness Function:** We must take the inverse of the total weight of the path because we are finding the shortest path between the two vertices.

For example, let us say that the fitness function is f(total). We used Dijkstra algorithm to find the shortest path.

Hence, f(total) = 1 / total, where total is the sum of all the weights for the shortest path.

**Expression:** We are considering each node as a gene. We are saving the edges in the Bag Data Structure. We have to do many searches, so we implemented Bag Data Structure.

**Cross-Over:** In the process of crossing-over, we select a random pair of genes and its index and randomly swap the values at that indices. What it will do is it won’t swap the entire set of genes, but some random pair of genes will be swapped to get the next set of generation which will have genes from both the previous genes.

For example,

Array1: |1|4|5|3|8|

After doing crossover at random indices and shuffle the Array1,

Array2: |5|3|2|4|1|

**Mutation:** In mutation, we do not perform things that we perform in a cross-over.

We just swap two random indices to interchange the genes.

For example,

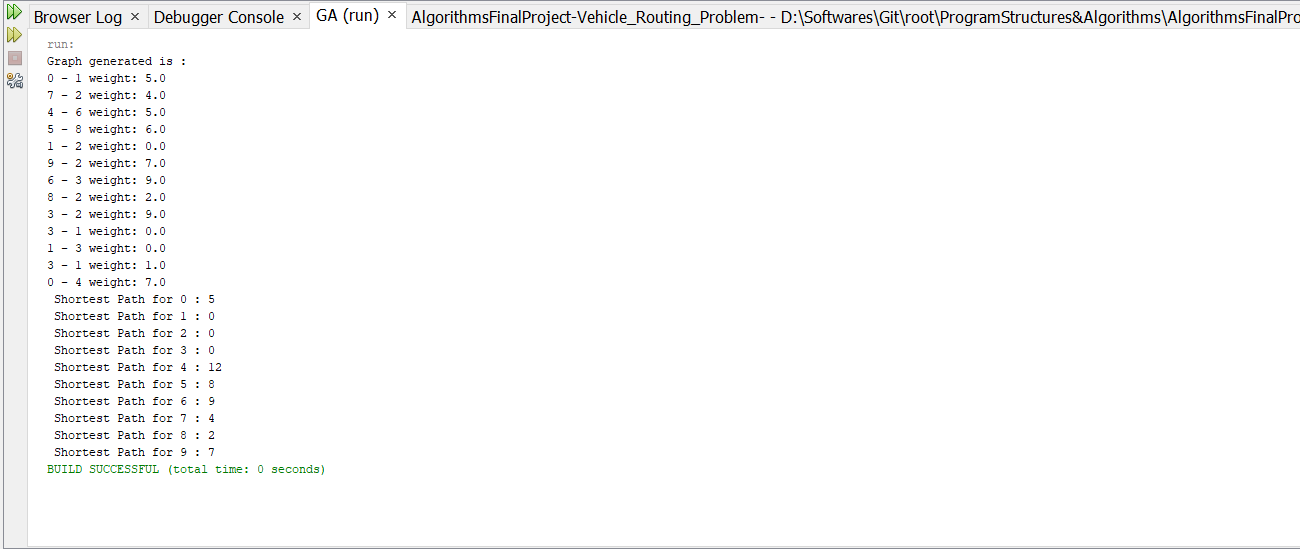
Array1: |2|4|6|8|1|3|5|7|9|

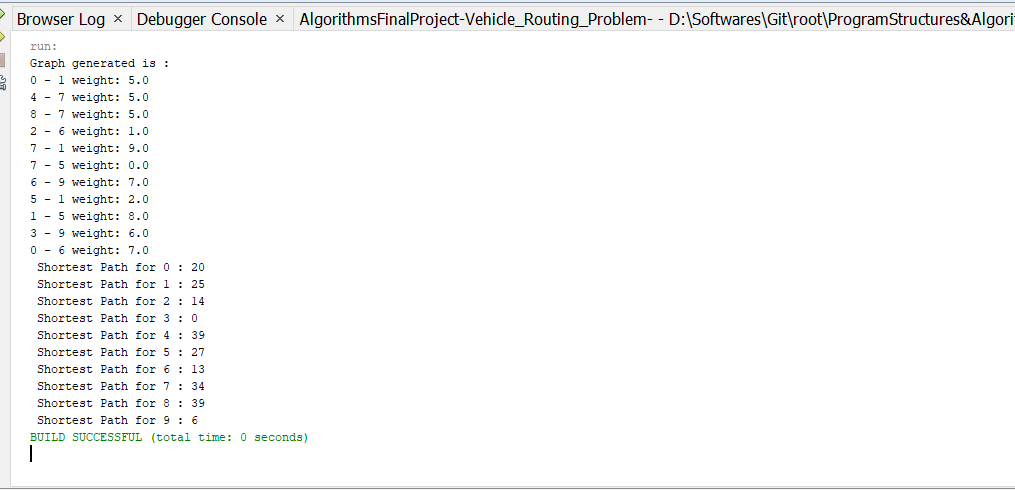
Swap the two random indices,

Array2: |2|3|6|8|1|4|5|7|9|

**Evolution**: In evolution, we will evolve the generations by finding the optimized solution from the solution set. For each generation, the population will double and the next generation will undergo mutation again. Then we calculated fitness function for each individual and sorted the values. So at last, after many generations the result will be a plateau.

**Screenshots** **of** **Results**:





**Graph-Generation:** We are creating a random edge weighted graph as our space in the one-dimensional plane.

