1. **Design of EA**
   A) Representation
      Route = a sequence of city ID's (chromosome)
      e.g. [2, 3, 5, 7, 1, 2, 10, …]

   B) Evaluation

      Route_length = distance(route) = $\Sigma\, cityDist(i, i+1)$

      Fitness could be evaluated in three ways:

      1. Favor and adapt the least one route_length
      2. 1/route_length
      3. E^(1/route_length)

   B) Crossover
      Multi-points cut / uniform cut
      Ensure each city in route appear just once after crossover
   C) Mutation
      Randomly swap two city ID's in route sequence
   D) Selection
      Tournament
2. **Technique to Advance EA**
   Crossover
   Cut on worst gene crossover – Given two parents find the worst gene, which would be the maximum distance from it's left neighbour, and then the two worst genes are compared, and the algorithm will get the worst of both. The index of this point is considered as a cut point in the parent that has the worst gene. This will eliminate the worst and increase fitness.
   Mutation
   Worst gene with worst gene mutation – search for the two worst genes and exchanged positions of both selected genes with each other. This will improve fitness after mutation by removing the least effective gene.
3. **Runtime Optimization**
   Place the dataset throughout the computation into a dictionary for increased runtime optimization. This creates optimization in the runtime by negating the need to traverse the entirety of an array in order to find a certain city.
   Multithreading on the data set. This will remove the need to traverse the dataset each time you are trying to do calculations. Multithreading will greatly shorten the runtime of the algorithm when we run large data sets.
4. **Team Management**
   Tingrui Hu – Graphing and visualization of data after computation
   Weiming Chen – Code for evolutionary algorithm
   Justin Heffernan – Final report and proposal
   *This is a rough distribution of the workload and team members will help in other areas when a member runs into any issues.