**Time Complexity Analysis:**

1. **Initialization of Population**: The initialization of the population involves creating **POP\_SIZE** permutations of **N** elements, which takes O(N \* POP\_SIZE) time.
2. **Main Loop (Generations)**:
   * **Fitness Calculation**: For each individual in the population, the fitness is calculated. This involves traversing the path of length **N**, so the time complexity is O(N \* POP\_SIZE).
   * **Selection, Crossover, and Mutation**: Within each generation loop, these operations are performed for each individual in the population.
     + Selection involves selecting two parents, which is O(1).
     + Crossover involves selecting a random portion of the parent's path, and copying it to the child. This operation has a time complexity of O(N).
     + Mutation involves swapping elements in the path with a probability of MUTATION\_RATE, which is O(N).
     + So, the total time complexity for these operations in each generation loop is O(N \* POP\_SIZE).
3. **Sorting Fitness Index**: Sorting the fitness index vector takes O(POP\_SIZE \* log(POP\_SIZE)) time.
4. **Overall**: Considering the above steps, the dominant factor in time complexity is the main loop (generations), which runs for **GENERATIONS** iterations. So, the overall time complexity of the algorithm is approximately O(GENERATIONS \* N \* POP\_SIZE \* log(POP\_SIZE)).

**Space Complexity Analysis:**

1. **Initialization of Population**: The population is represented as a 2D vector of size POP\_SIZE \* N. So, the space complexity is O(N \* POP\_SIZE).
2. **Fitness Index Vector**: The fitness index vector stores pairs of doubles and integers for each individual in the population, which is O(POP\_SIZE).
3. **New Population Vector**: The new population vector is similar to the population vector, so its space complexity is also O(N \* POP\_SIZE).
4. **Overall**: The space complexity is dominated by the population and new population vectors, so the overall space complexity is O(N \* POP\_SIZE).

**Summary:**

* Time Complexity: O(GENERATIONS \* N \* POP\_SIZE \* log(POP\_SIZE))
* Space Complexity: O(N \* POP\_SIZE)

***NORMALLY THE TIME COMPLEXITY OF THIS PROBLEM USING GENETIC ALGORITHM IS O(n!)***

***SPACE COMPLEXITY IS O(n)***