Steve Dixon

4560 HW 2

2 October 2020

**N Queens:**

Representation:

Each queen was an object that knew its x and y position. An N queen individual or solution candidate was an array of length N queens. The population was an array of NQ individuals. The NQ individuals were aware of duplicate queens in their genotype and would randomly produce another queen to put in its place.

Fitness was calculated as 1/number of collisions unless the number of collisions was zero then the fitness was assigned the maximum possible integer value (2^32)-1.

parenthood selection:

For parenthood selection, I used a tournament of 4 contestants and 2 parents were selected. The parent’s fitness was multiplied by a random multiplier of 1, 2, or 3. The constant with the higher fitness combined with the multiplier was selected to be a parent. The multiplier was created to give weaker solutions a chance to mate. A weak candidate may get lucky and get a high multiplier and beat its competitor’s fitness.

Mutation

During crossover, if queens switched positions and that queen was already in the DNA of the child then an RNG would choose between 0-3. Based on the number selected the parent 1 queen x value, parent 2 queen x value, parent 1 queen y value, or parent 2 queen y value would mutate by changing to a randomly selected value between 0 and N.

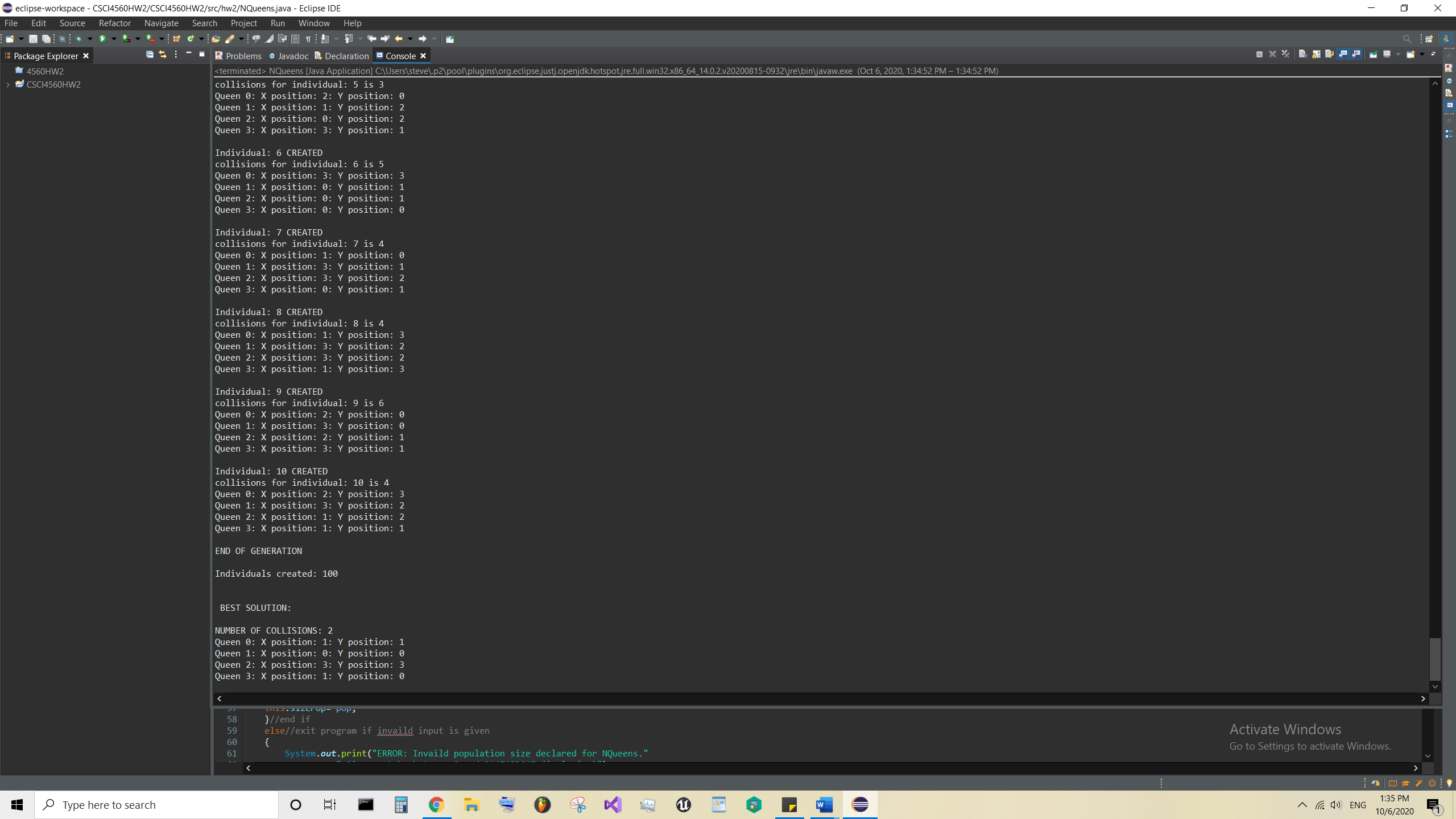
Crossover

For crossover I used a random number generator to select 0 or 1 over the DNA indices. If the RNG landed on 1, the queen x and y positions of the parents switched. If the RNG landed on zero, the index incremented, and no queens switched positions. This uniform crossover resulted in two children that were added to the next generation.

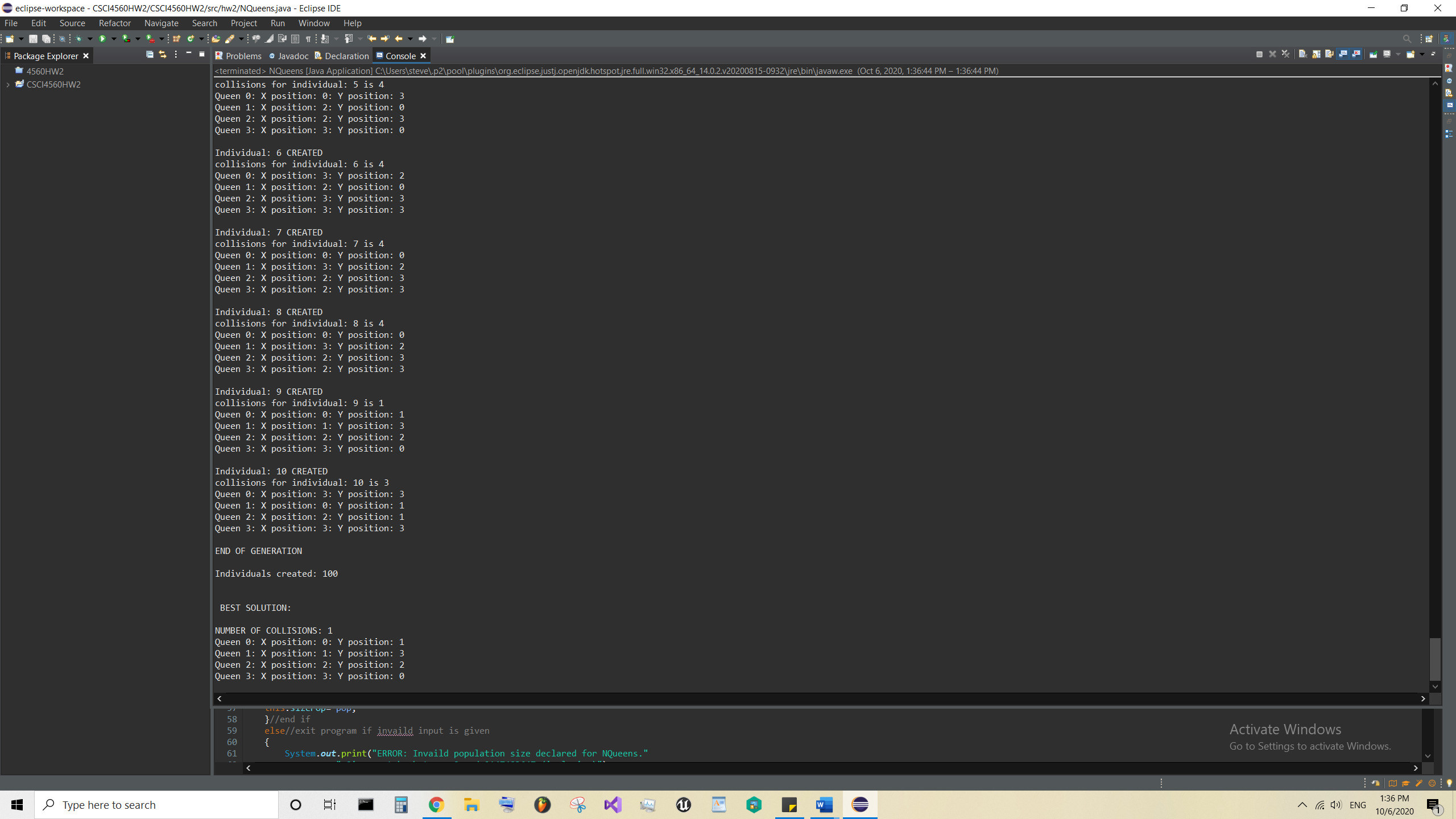
survival selection

All individuals are killed after the end of a generation. A new generation is completed once enough children have been created to replace the older generation. Until then, each individual has some chance of procreation.

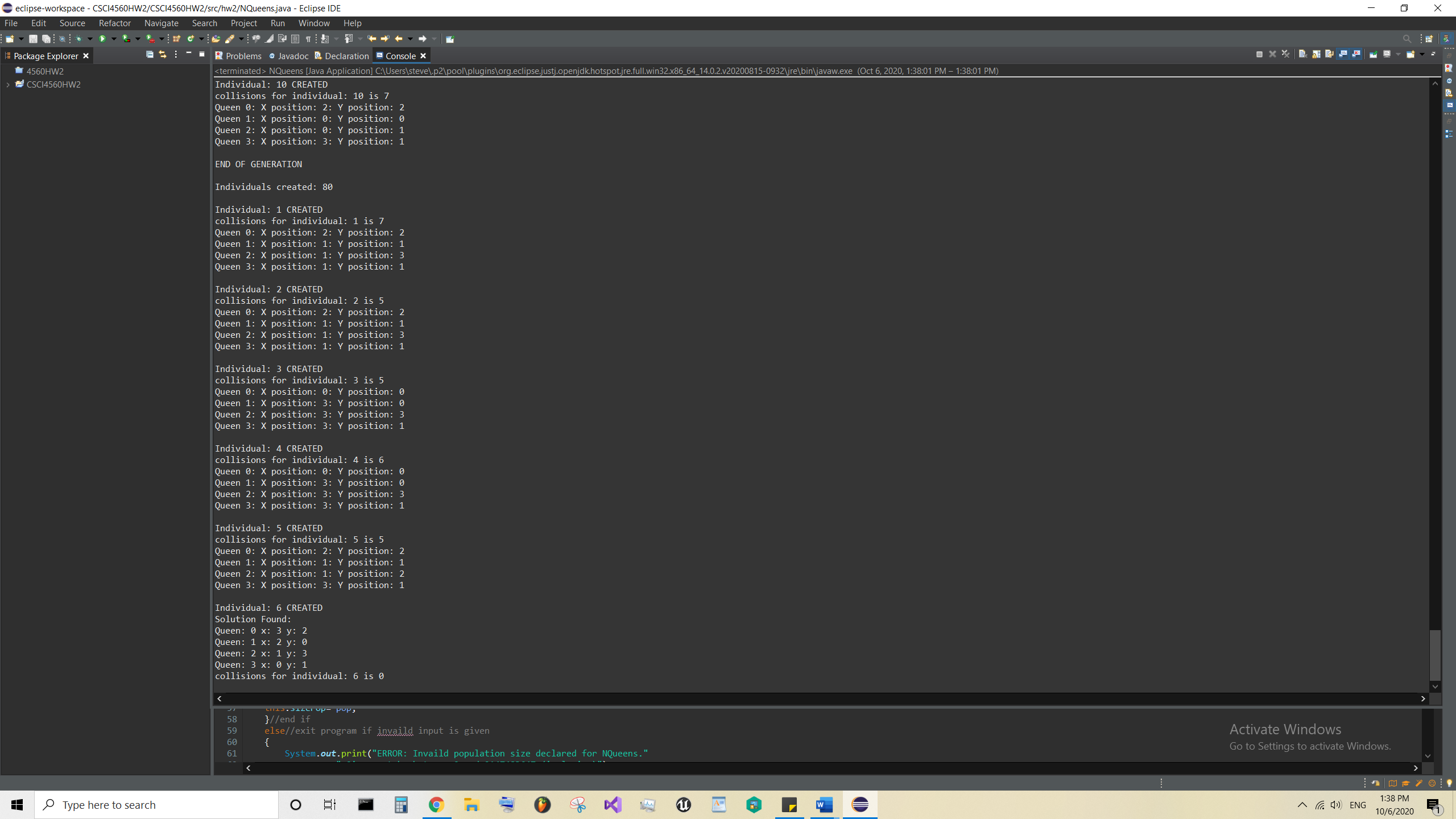
Run 1



Run 2



**Example of solution found**



**Traveling Salesmen:**

Representation

Each position was represented by a node object with an x and y value. An individual’s DNA was an array of these nodes. The population was an array of these individuals.

The fitness function for this problem was the known minimum distance (given in HW description) divided by the total distance traveled by an individual.

Parenthood selection

For parenthood selection, I used a tournament of 4 contestants and 2 parents were selected. The parent’s fitness was multiplied by a random multiplier of 1, 2, or 3. The constant with the higher fitness combined with the multiplier was selected to be a parent. The multiplier was created to give weaker solutions a chance to mate. A weak candidate may get lucky and get a high multiplier and beat its competitor’s fitness.

Mutation

I did not use mutation.

Crossover

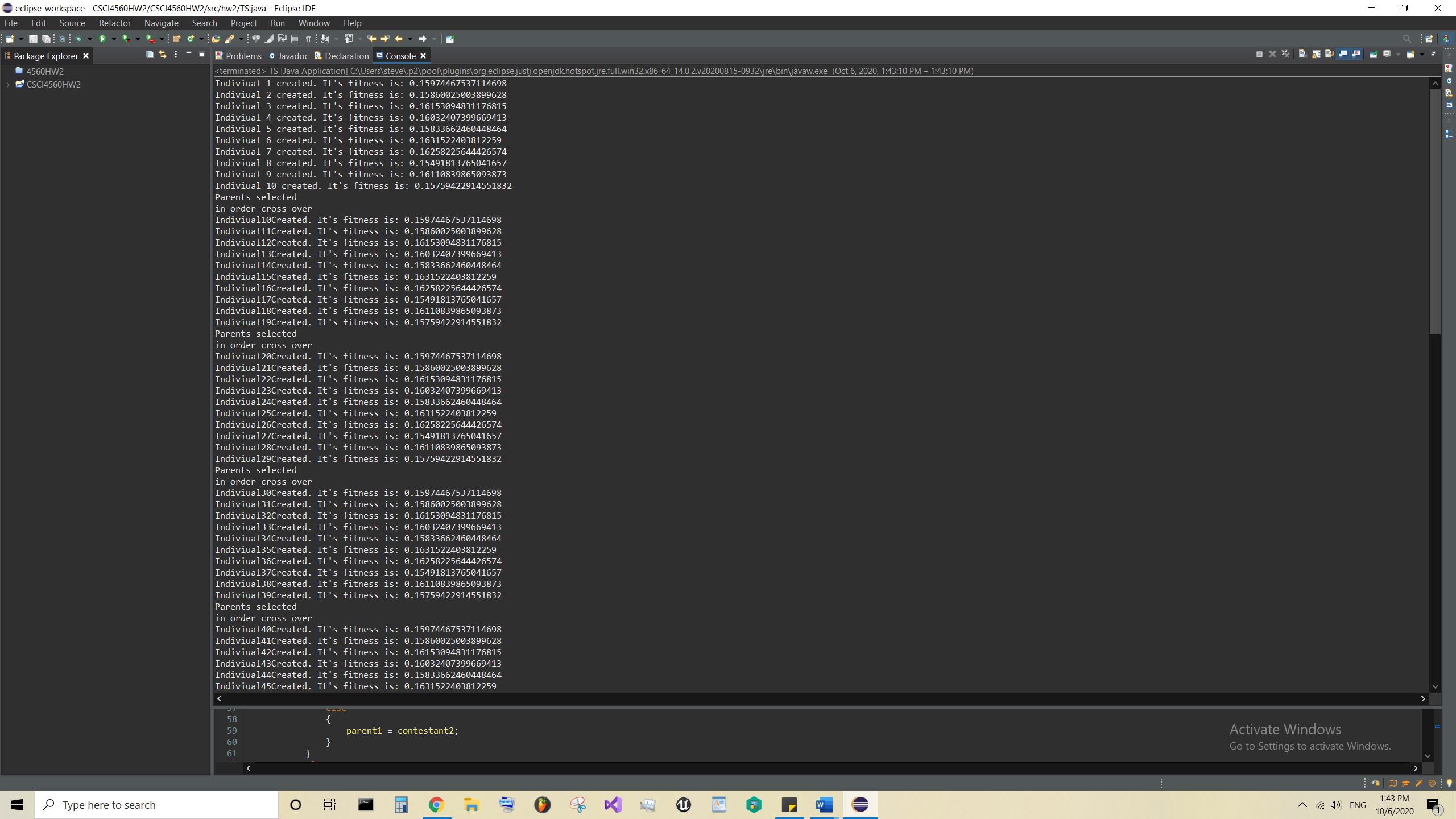
I chose to use order crossover for this problem because each city must be traveled to once therefore the cities cannot be randomly crossed over. This would cause missing cited, redundant cities, or cities not initially defined.

survival selection

All individuals are killed after the end of a generation. A new generation is completed once enough children have been created to replace the older generation. Until then, each individual has some chance of procreation.

Screenshots:

Run 1



Run 2