**N-Queens Problem Using Genetic Algorithm**

**Genetic Algorithm**

* Genetic Algorithm are non-deterministic algorithm that follows a natural order to determine the best possible solution of a problem. In our program the genetic algorithm will be used in order to determine the conflicts between any two queens on the chess board and will assign the fitness of the pattern according to the conflict.
* The fittest patterns along with their offspring’s will pass on to the next generation and conflicts will be calculated accordingly. To eliminate any possibility of biased result, we have added mutations in our code so that a few mutated patterns are added after fixed intervals. This would ensure that the results that we get are not biased to the parents that are generated.

### Problem Statement

* The N-Queens problem is a classic problem that is used to determine the position of queens that can be spread on the chess board such that no two queens can kill each other. The problem can be solved in many possible ways including backtracking, branch and bound, etc.  
  Here in our project we try to solve N-Queens using Genetic algorithm. The general running time of N queens problem is O(n!) and hence we have tried to reduce this using our genetic algorithm approach.

### Approach

### We followed the approach of solving the N-Queens by implementing chromosomes which have maximum length, fitness, conflicts.

### Firstly, we plotted the queens on the diagonals and then checked all four directions to make sure that there are no conflicts. The fitness is calculated based on the conflicts.

### Based on the fitness, parent with highest fitness will come first, followed by the next in order and so on.

### For mating, we have used cross over partial mapping.

### Mutation: If the queen in the parent has same position in the two boards, same position will be given to the children in the new board. But if the position of parents in both boards are different, child would be given a random value in the new board.

### Using Roulette wheel for selecting parent starting from a fixed point and then spinning the wheel. The region which comes in the front of the fixed point is chosen as the parent. And the process is repeated for the next iteration.

### Genotype in genetic algorithm corresponds to chromosome in our case(heritable aspect)

### Phenotype is defined as the number of conflicts between queens (the trait by which two queens can attack each other).

### Fitness which is calculated based on the number of conflicts of the queens. So if there are maximum conflicts, fitness will be lowest and vice versa.

### Our test cases include test cases for fitness, calculation of offspring’s.

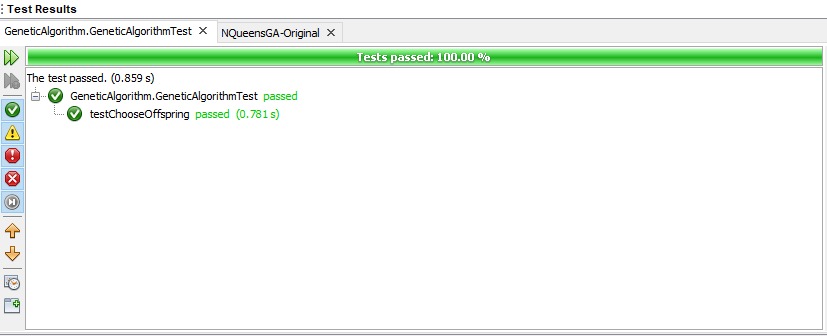
**Implementation Details:**

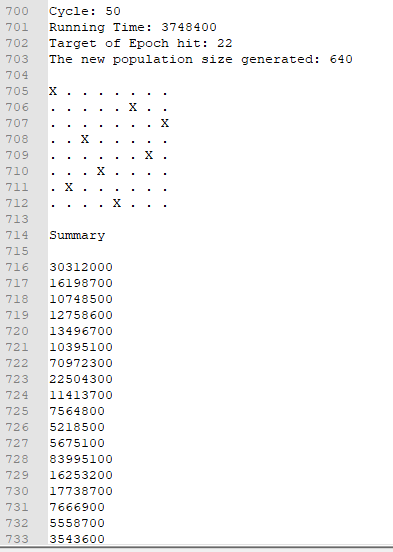
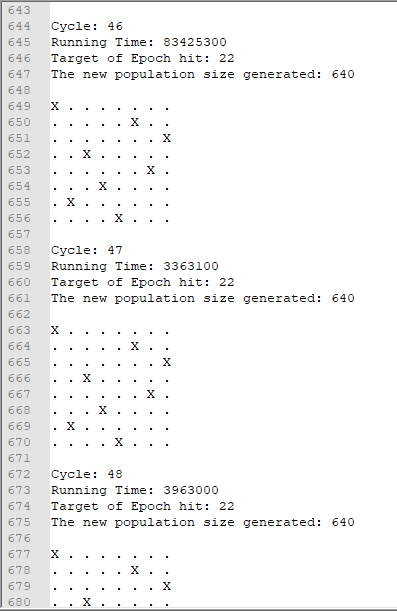
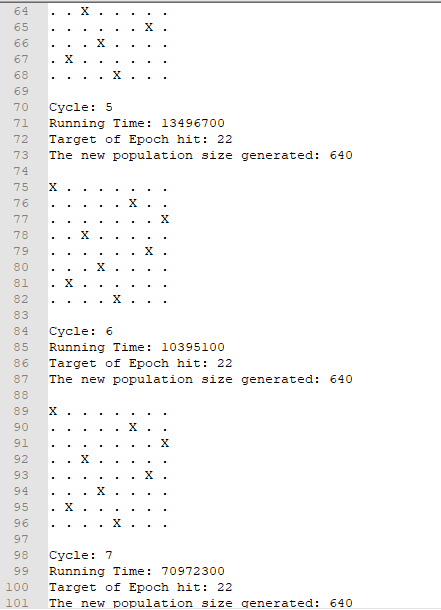
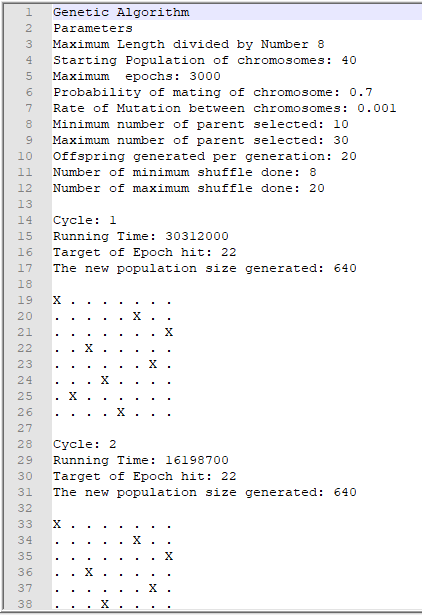
1. We have implemented genetic algorithm separately which encompasses the genetic code and also generates the random mutation.
2. **Gene expression:** We had three fixed criteria to find the fitness function. The position of the queen should be such that no other queen is present horizontally, vertically and diagonally. Such a criteria enabled us to find the number of conflicts (fitness function).
3. **Fitness function:** The above criteria enabled us to find the number of conflicts. The more the number of conflicts, fitness will be the lowest and vice versa.

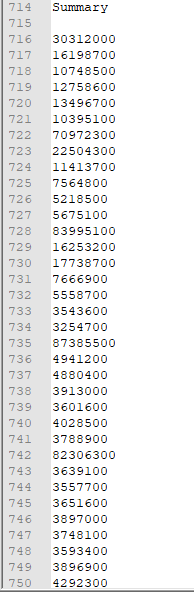
**Results:**

**Unit Test Cases:**

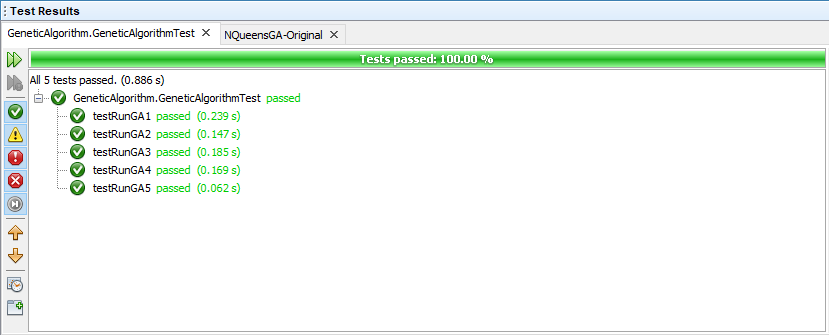
Test cases for offspring’s:

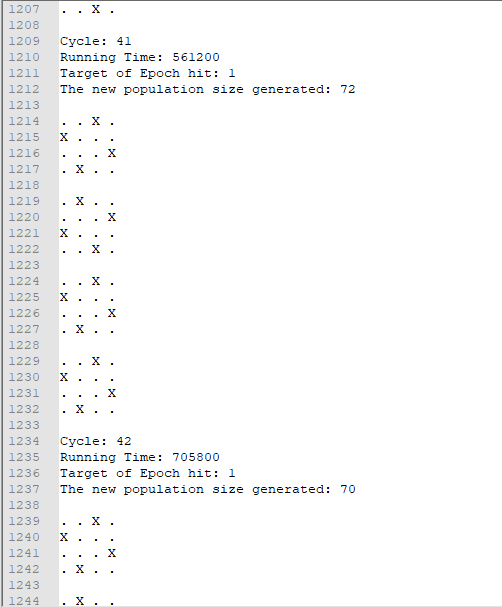
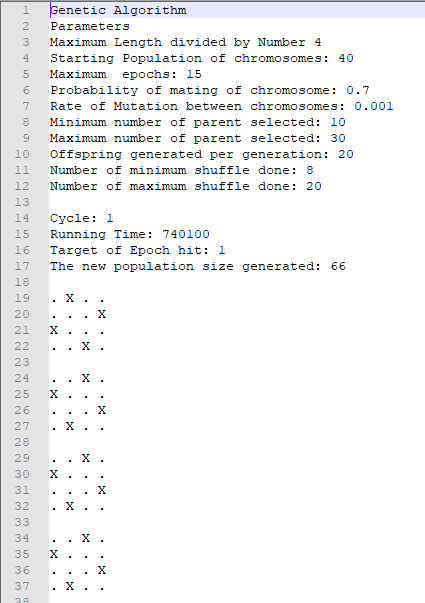


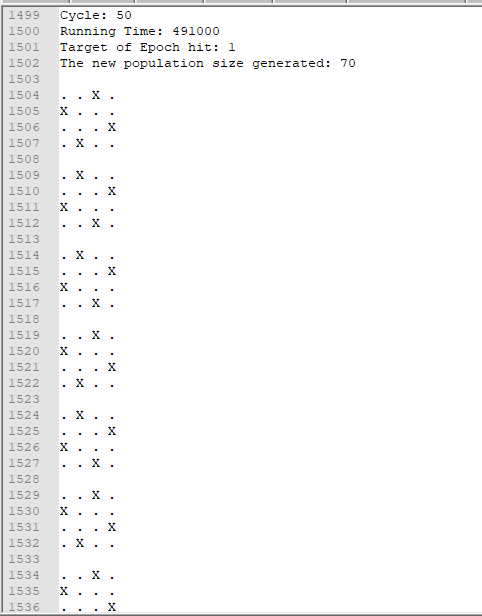


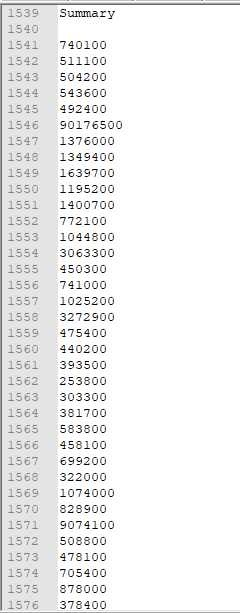


Test cases results when running genetic algorithm:

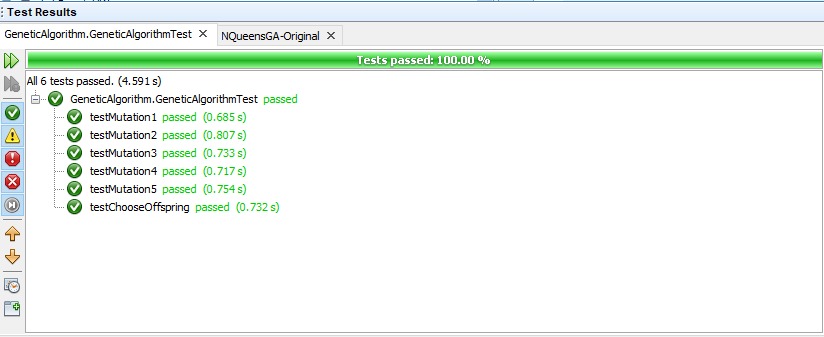


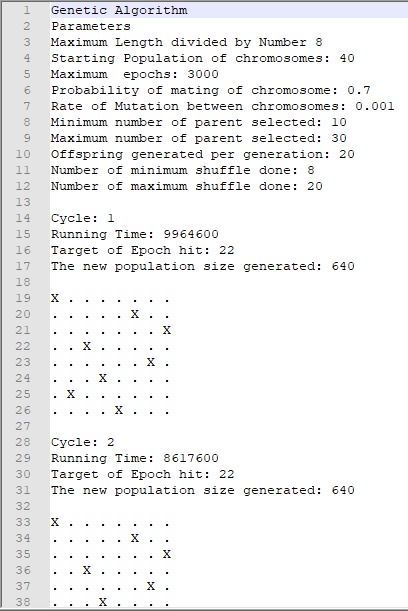


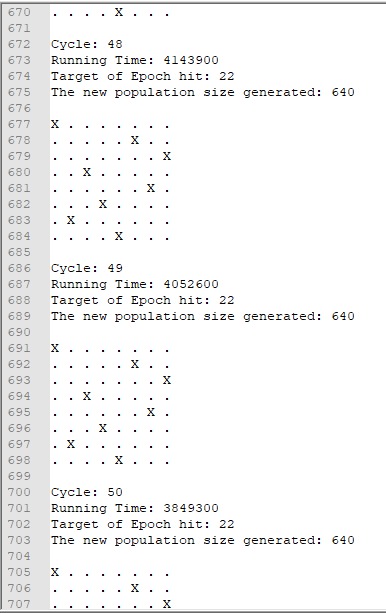


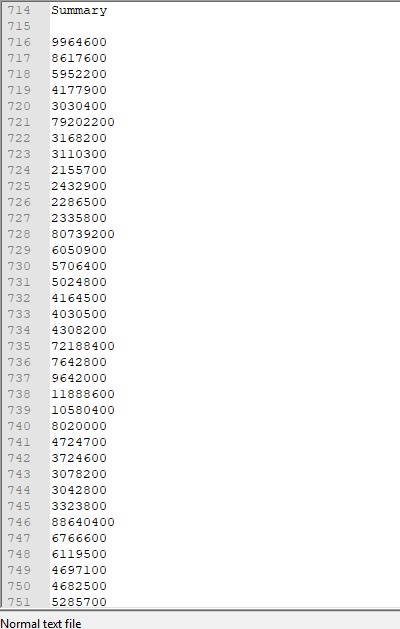


Test cases for mutation:









**Conclusion:**

In this we have enacted the N queens problem using genetic algorithm, a non-deterministic algorithm and have documented our results.

**References:**

**Genetic Algorithm:**  
[Genetic Algorithm Introdcution](https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3)  
[Creating a Genetic Algorithm Project](http://www.theprojectspot.com/tutorial-post/creating-a-genetic-algorithm-for-beginners/3)  
[Tutorial on Genetic Algorithm](https://www.tutorialspoint.com/genetic_algorithms/index.htm)

**N queens Algorithm:**  
[Understanding N queens Using Backtracking Algorithm](https://www.geeksforgeeks.org/n-queen-problem-backtracking-3/)  
[Improving N queens using Branch and Bound Method](https://www.geeksforgeeks.org/n-queen-problem-using-branch-and-bound/)  
[Java Code for N queens Algorithm Using Genetic Algorithm](http://mnemstudio.org/ai/ga/nqueens_java_ex1.txt)