Genetic Algorithm for 8 Queens

Name: Sree Vandana Nadipalli

Genetic Algorithm technique:

1. ***Initialization***: select 'k' number of random individuals indicating placement of queens in a column on a chess board. In the program the position of the queen is represented using string of 8 digit numbers between the range [1, 8]. Example: one individual is "24748572". Individuals are stored in population list.

2. ***Fitness Function***: The fitness function used here is "total number of non-attacking queens for a given individual". Greater the fitness value better is the solution. The fitness function values fall within the range [0, 28] where 28 is the highest best score.
   In the program, the total number of non-attacking queens' value is calculated like (28 – total number of attacks).
   Total number of attacking queens is calculated using 3 attacking functions, which calculated attacks in each direction.
   Attack's calculation functions:
   I. *rowColAttacks*: calculates number of queens attacking horizontally and vertically.
   II. *diagonalAttack45*: number of queens attacking in 45-degree diagonal on board.
   III. *diagonalAttack135*: number of queens attacking in 135-degree diagonal on board.

3. ***Selection:*** Tow individuals from the population are selected to produce the next generation. One individual can be selected multiple time or not even single time. The selection happenes k number of times, so that the new generation generated is the same length of initial population.

4. ***Cross Over:*** A random index 'ri' is selected and queen locations from 0 to 'ri' from parent 1 and from 'ri+1' to 8 from second parent are selected to form a new child of next generation.

5. ***Mutation:*** The probability of mutation happening is 0.5 . A function "smallRandomProbability" is written to determine if the mutation should happen on the child or not. In a given set of {0, 1} the probability of selecting any one of them is 0.5, so this function randomly select between 0 and 1, if 1 is selected mutation happens and if 0 is selected mutation does not happened on the child.

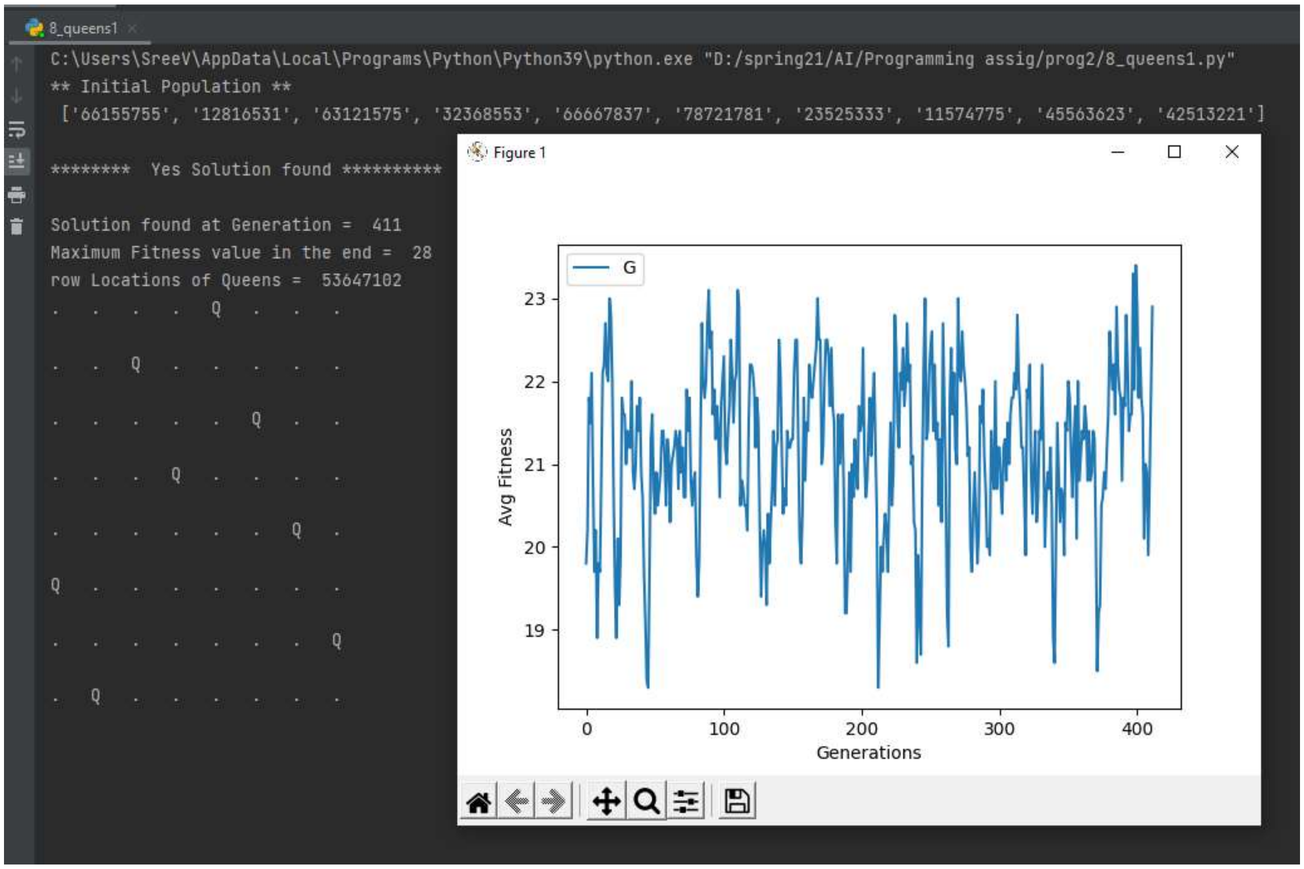
6. ***Iteration:*** Iterated from step 2 to 5 till the termination condition is met. Here the termination condition is, either "28 is present in the fitness values list calculated for a population" or "till maximum number of iterations are reached".

For writeup including the graph and the maximum fitness, Generation at which the solution is found and final solution board with queens' location in string format are displayed below.
All generations along with maximum fitness of the generation and average fitness and Individual with highest fitness in that generation can be seen when executing the program.

How to execute the program is written on top of GA program file.

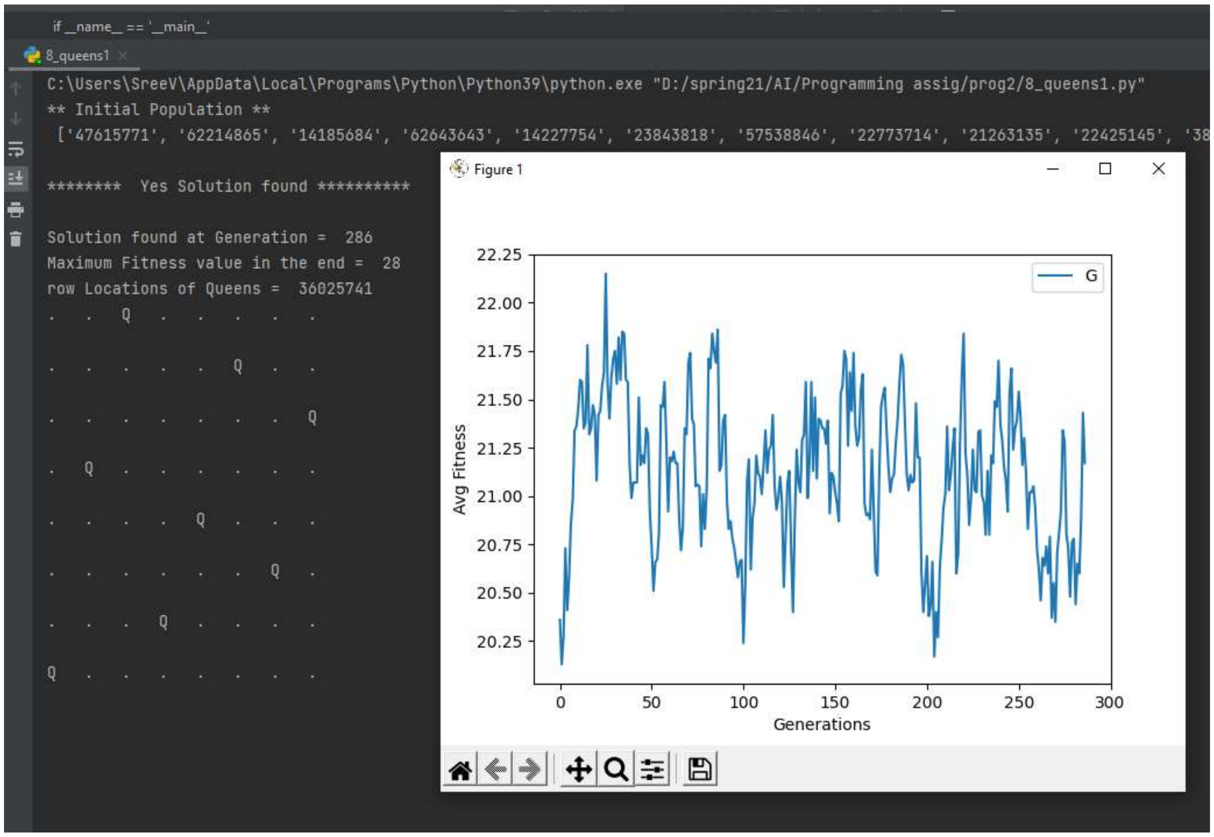
Program output:

1. ***For population size = 10 and Maximum Iteration = 1000***



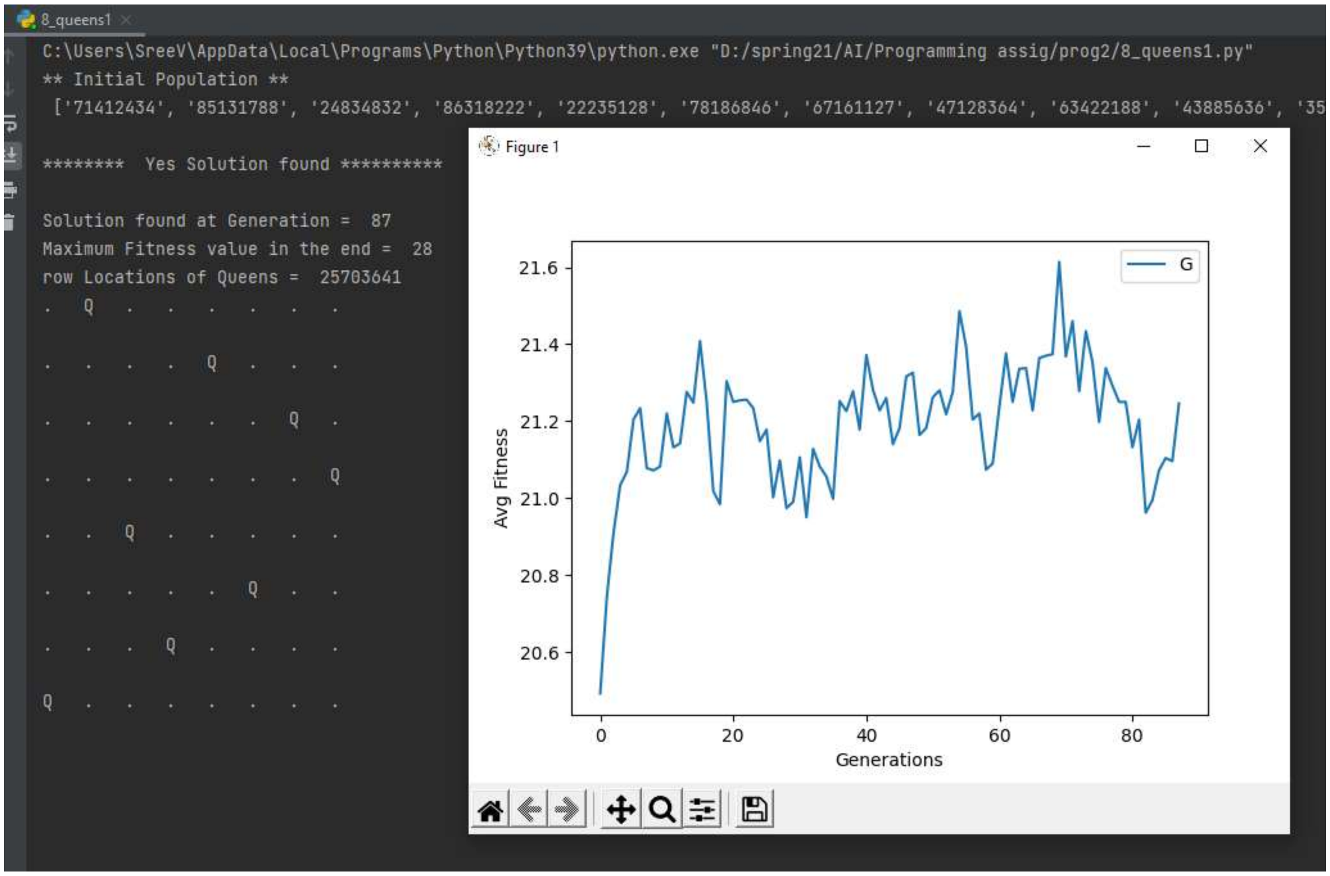
```
C:\Users\SreeV\AppData\Local\Programs\Python\Python39\python.exe "D:/spring21/AI/Programming assig/prog2/8_queens1.py"
** Initial Population **
 ['66155755', '12816531', '63121575', '32368553', '66667837', '78721781', '23525333', '11574775', '45563623', '42513221']

********  Yes Solution found **********

Solution found at Generation =  411
Maximum Fitness value in the end =  28
row Locations of Queens =  53647102
.   .   .   .   Q   .   .   .

.   .   Q   .   .   .   .   .

.   .   .   .   .   Q   .   .

.   .   .   Q   .   .   .   .

.   .   .   .   .   .   Q   .

Q   .   .   .   .   .   .   .

.   .   .   .   .   .   .   Q

.   Q   .   .   .   .   .   .
```

2. ***For population size = 100 and Maximum Iteration = 1000***



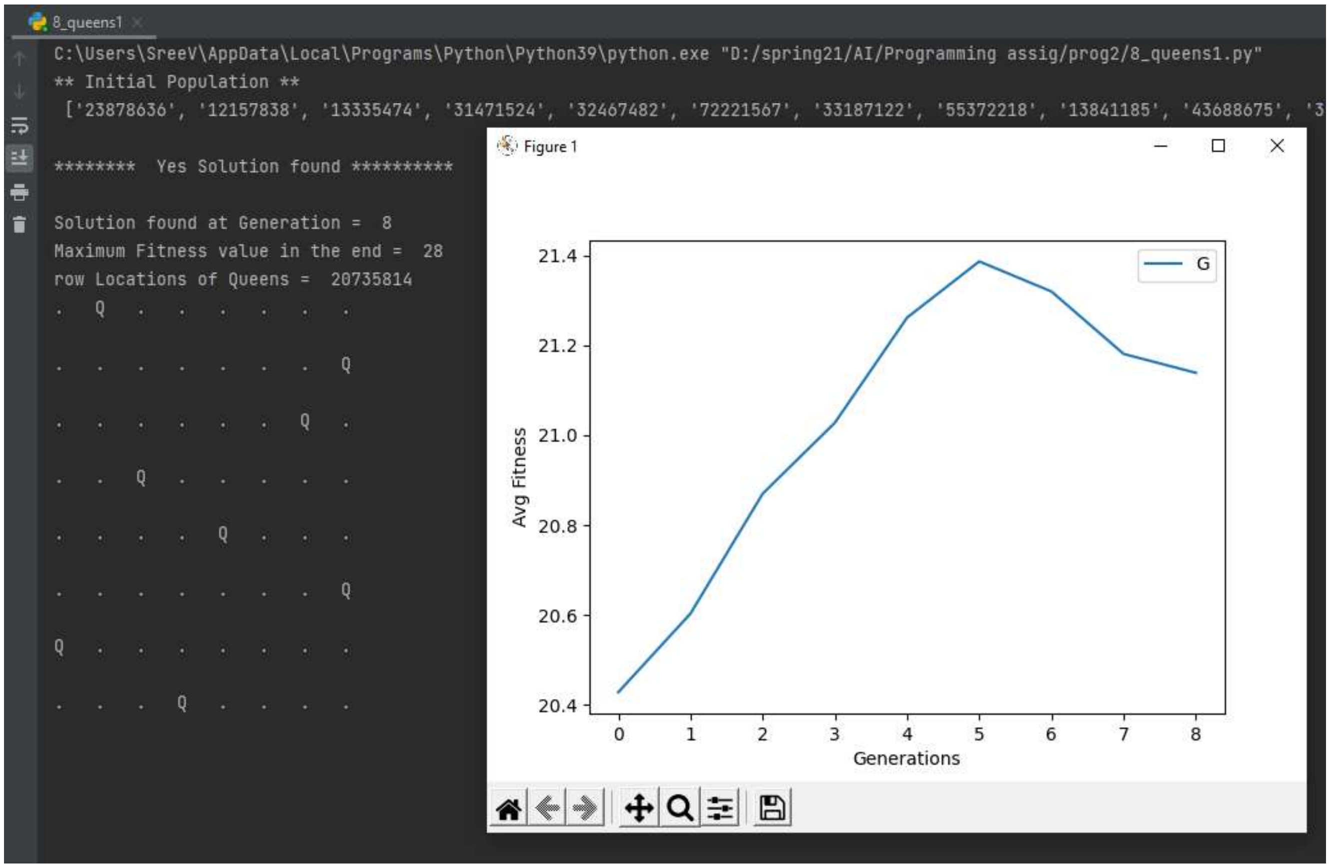
```
C:\Users\SreeV\AppData\Local\Programs\Python\Python39\python.exe "D:/spring21/AI/Programming assig/prog2/8_queens1.py"
** Initial Population **
 ['47615771', '62214865', '14185684', '62643643', '14227754', '23843818', '57538846', '22773714', '21263135', '22425145', '38

********  Yes Solution found **********

Solution found at Generation =  286
Maximum Fitness value in the end =  28
row Locations of Queens =  36025741
.   .   Q   .   .   .   .   .

.   .   .   .   .   Q   .   .

.   .   .   .   .   .   .   Q

.   Q   .   .   .   .   .   .

.   .   .   .   Q   .   .   .

.   .   .   .   .   .   Q   .

.   .   .   Q   .   .   .   .

Q   .   .   .   .   .   .   .
```

3. ***For population size = 500 and Maximum Iteration = 1000***



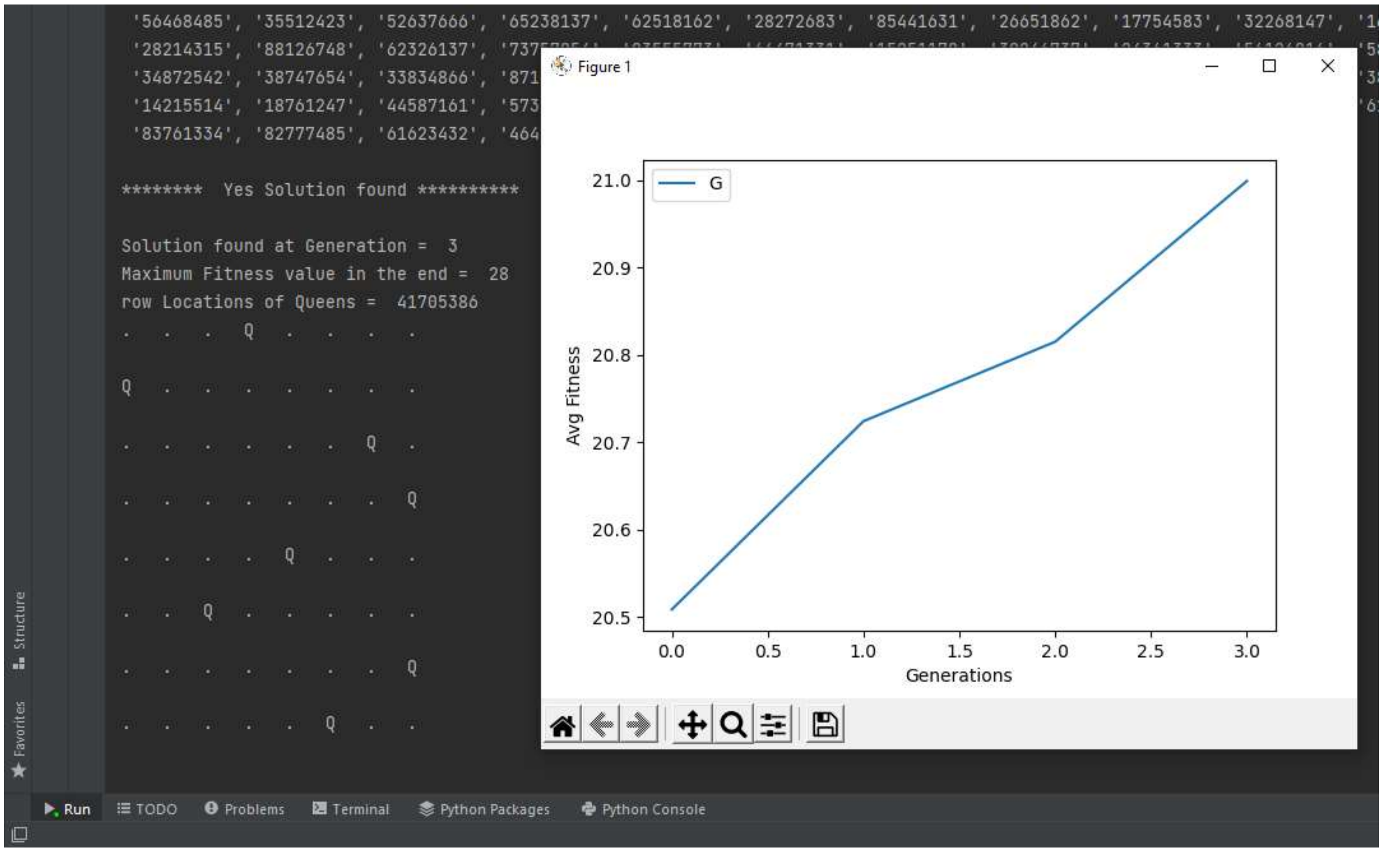
```
C:\Users\SreeV\AppData\Local\Programs\Python\Python39\python.exe "D:/spring21/AI/Programming assig/prog2/8_queens1.py"
** Initial Population **
 ['71412434', '85131788', '24834832', '86318222', '22235128', '78186846', '67161127', '47128364', '63422188', '43885636', '35

********  Yes Solution found **********

Solution found at Generation =  87
Maximum Fitness value in the end =  28
row Locations of Queens =  25703641
.   Q   .   .   .   .   .   .

.   .   .   .   .   Q   .   .

.   .   .   .   .   .   .   Q   .

.   .   .   .   .   .   .   .   Q

.   .   Q   .   .   .   .   .

.   .   .   .   .   .   Q   .   .

.   .   .   Q   .   .   .   .

Q   .   .   .   .   .   .   .
```

4. ***For population size = 1000 and Maximum Iteration = 1000***



```
C:\Users\SreeV\AppData\Local\Programs\Python\Python39\python.exe "D:/spring21/AI/Programming assig/prog2/8_queens1.py"
** Initial Population **
 ['23878636', '12157838', '13335474', '31471524', '32467482', '72221567', '33187122', '55372218', '13841185', '43688675', '3

********  Yes Solution found **********

Solution found at Generation =  8
Maximum Fitness value in the end =  28
row Locations of Queens =  20735814
.   Q   .   .   .   .   .   .

.   .   .   .   .   .   .   .   Q

.   .   .   .   .   .   .   Q   .

.   .   Q   .   .   .   .   .

.   .   .   .   .   Q   .   .   .

.   .   .   .   .   .   .   .   Q

Q   .   .   .   .   .   .   .

.   .   .   Q   .   .   .   .
```

5. ***For population size = 10000 and Maximum Iteration = 1000***



```
 '56468485', '35512423', '52637666', '65238137', '62518162', '28272683', '85441631', '26651862', '17754583', '32268147', '1
 '28214315', '88126748', '62326137', '737                                                                                  '5
 '34872542', '38747654', '33834866', '871                                                                                  '3
 '14215514', '18761247', '44587161', '573                                                                                  '6
 '83761334', '82777485', '61623432', '464

********  Yes Solution found **********

Solution found at Generation =  3
Maximum Fitness value in the end =  28
row Locations of Queens =  41705386
.   .   .   Q   .   .   .   .

Q   .   .   .   .   .   .   .

.   .   .   .   .   .   Q   .

.   .   .   .   .   .   .   Q

.   .   .   .   Q   .   .   .

.   .   Q   .   .   .   .   .

.   .   .   .   .   .   .   Q

.   .   .   .   .   Q   .   .
```

<u>Program Findings and Conclusions:</u>

Population size	Iteration at which solution is found
10	411
100	286
500	87
1000	8
10,000	3

We can clearly see that as population size increased, the number of Iterations during which the solution is found got reduced. However due to increase in population size the execution time increased due to increase in internal loops. Even though the time took to run 10,000 population size is just few seconds in this case, in general, for running genetic algorithm with highest population size in real world situations might increase drastically.