ASSIGNMENT (PART-A)

AADHARSH K XAVIER (B200730CS)

1. 4-Queens Problem

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

How many queens are fixed? (0-3):  1
Enter fixed queens:
0 1

INITIAL CONFIGURATION:
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]


STATE SPACE CONFIGURATIONS:
       [(1, 3)]
       [(1, 3)]
       [(1, 3)]
       [(1, 3)]
    [(1, 3), (3, 2)]
       [(3, 3)]
       [(3, 3)]
       [(0, 3)]
       [(0, 3)]
    [(3, 3), (0, 2)]
       [(3, 3)]
       [(3, 3)]
       [(0, 3)]
       [(1, 3)]
    [(1, 3), (3, 2)]
       [(1, 3)]
       [(1, 3)]
       [(1, 3)]
       [(1, 3)]
    [(1, 3), (2, 2)]
  [(1, 3), (3, 2), (2, 0)]

OPTIMAL CONFIGURATION:
[0, 1, 0, 0]
[0, 0, 0, 1]
[1, 0, 0, 0]
[0, 0, 1, 0]
```

- The rows are numbered 0 to 3 from top to bottom and the columns 0 to 3 from left to right.
- In the above image, one queen is fixed at (0, 1). The remaining 3 queens are placed as shown in the optimal configuration. The algorithm **recursively** checks every position for a queen in each column and selects the one with minimum conflicts with other queens.
- In the above state space configurations, the first four lines represent the minimum conflict positions for the 4th queen when the 3rd queen is at (0, 2), (1, 2), (2, 2) and (3, 2) respectively.

Similarly, the fifth line [(1, 3), (3, 2)] represents the minimum conflict positions for the 3rd and 4th queens when the 1st queen is at cell (0, 0) *(It is not the 2nd queen's position since its position is already fixed by user).*

- The last line [(1, 3), (3, 2), (2, 0)] represents the optimal positions for all the remaining three queens.

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS

How many queens are fixed? (0-3):  1
Enter fixed queens:
0 0

INITIAL CONFIGURATION:
[1, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]


STATE SPACE CONFIGURATIONS:
      [(1, 3)]
      [(1, 3)]
      [(1, 3)]
      [(1, 3)]
    [(1, 3), (3, 2)]
      [(2, 3)]
      [(2, 3)]
      [(0, 3)]
      [(0, 3)]
    [(0, 3), (3, 2)]
      [(1, 3)]
      [(1, 3)]
      [(1, 3)]
      [(1, 3)]
    [(1, 3), (3, 2)]
      [(2, 3)]
      [(2, 3)]
      [(0, 3)]
      [(0, 3)]
    [(2, 3), (0, 2)]
  [(1, 3), (3, 2), (0, 1)]

OPTIMAL CONFIGURATION:
[1, 1, 0, 0]
[0, 0, 0, 1]
[0, 0, 0, 0]
[0, 0, 1, 0]
```

- In the above screenshot, the position of the 1st queen is fixed at (0, 0). This would not give a solution with 0 attacking pairs. However, the algorithm outputs the **optimal configuration** with just **one attacking pair** as shown.

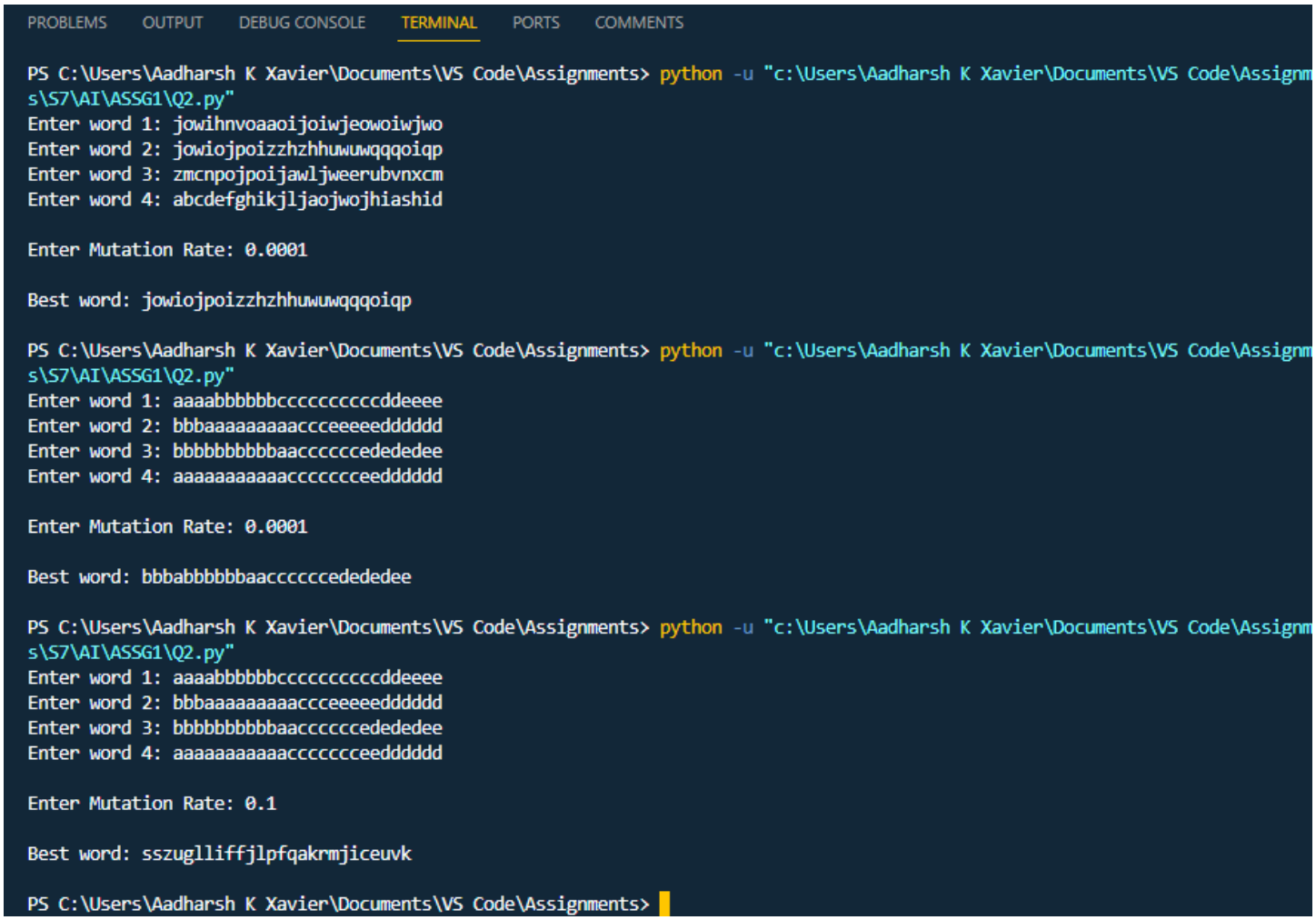
```
PS C:\Users\Aadharsh K Xavier\Documents\VS Code\Assignments> python -u "c:\Users\Aadharsh K Xavier\Documents\VS Code\Assignments\S7\AI\ASSG1\Q1.py"
How many queens are fixed? (0-3):  2
Enter fixed queens:
1 1
2 2

INITIAL CONFIGURATION:
[0, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 0]


STATE SPACE CONFIGURATIONS:
       [(0, 3)]
       [(0, 3)]
       [(0, 3)]
       [(0, 3)]
     [(0, 3), (1, 0)]

OPTIMAL CONFIGURATION:
[0, 0, 0, 1]
[1, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 0]
```

- Here the 2nd and the 3rd queens are fixed at cells (1, 1) and (2, 2) respectively. The algorithm finds the optimal configuration with just **two attacking pairs.**

```
PS C:\Users\Aadharsh K Xavier\Documents\VS Code\Assignments> python -u "c:\Users\Aadharsh K Xavier\Documents\VS Code\Assignments\S7\AI\ASSG1\Q1.py"
How many queens are fixed? (0-3):  3
Enter fixed queens:
1 0
0 1
3 2

INITIAL CONFIGURATION:
[0, 1, 0, 0]
[1, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 1, 0]


STATE SPACE CONFIGURATIONS:
       [(0, 3)]

OPTIMAL CONFIGURATION:
[0, 1, 0, 1]
[1, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 1, 0]
```

- Here the positions of all queens other than the 4th queen are fixed by the user as shown in the initial configuration. The optimal position for the 4th queen is shown in the optimal configuration with **two attacking pairs.**

2. Genetic Algorithm



- In the first example in the above image, the output shows the **best-fit** word after running the genetic algorithm on the 4 random words.

- The second example shows the output when a very small mutation rate of 0.0001 is used and the third example shows the output when a bigger mutation rate of 0.1 is used on the same 4 words. This shows how the rate of mutation can make the children very different from their ancestors.