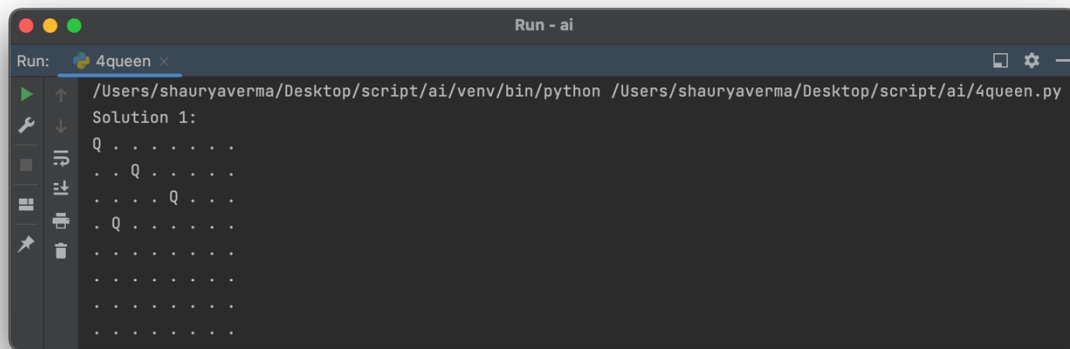
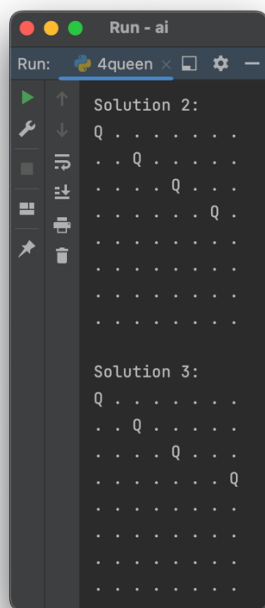


This solves the problem of 4queens using Breadth-First Search Algorithm.

Problem - Placing 4 queens on a standard chessboard so that no two queens attack each other.

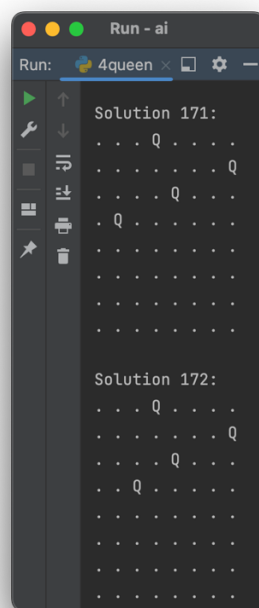


```
Run - ai
Run: 4queen x
/Users/shauryaverma/Desktop/script/ai/venv/bin/python /Users/shauryaverma/Desktop/script/ai/4queen.py
Solution 1:
Q . . . . .
. Q . . . .
. . . Q . .
. Q . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```



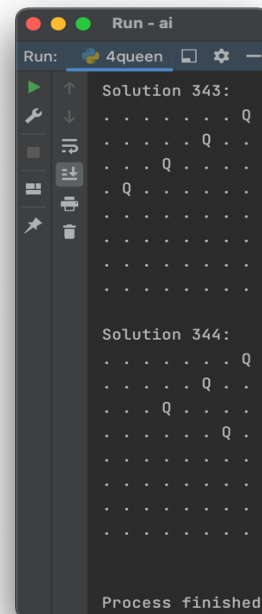
```
Run - ai
Run: 4queen x
Solution 2:
Q . . . . .
. Q . . . .
. . . Q . .
. Q . . . .
. . . . .
. . . . .
. . . . .
. . . . .

Solution 3:
Q . . . . .
. Q . . . .
. . . Q . .
. . . . Q
. . . . .
. . . . .
. . . . .
. . . . .
```



```
Run - ai
Run: 4queen x
Solution 171:
. . . Q . .
. . . . Q
. . . . Q
. . . Q . .
. Q . . . .
. . . . .
. . . . .
. . . . .

Solution 172:
. . . Q . .
. . . . Q
. . . Q . .
. . Q . . .
. . . . .
. . . . .
. . . . .
. . . . .
```



```
Run - ai
Run: 4queen x
Solution 343:
. . . . . Q
. . . . Q .
. . . Q . .
. Q . . . .
. . . . .
. . . . .
. . . . .
. . . . .

Solution 344:
. . . . . Q
. . . . Q .
. . . Q . .
. . . . Q .
. . . . .
. . . . .
. . . . .
. . . . .

Process finished
```

Total – 344 solutions

Checkout the pygame repository for a graphical representation of the same.

Happy Coding!