RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM, CHENNAI 602105

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

AI19341

PRINCIPLES OF ARTIFICIAL INTELLIGENCE

THIRD YEAR

FIFTH SEMESTER

INDEX

EX.NO: 1 **DATE: 09 - 08 - 2024**

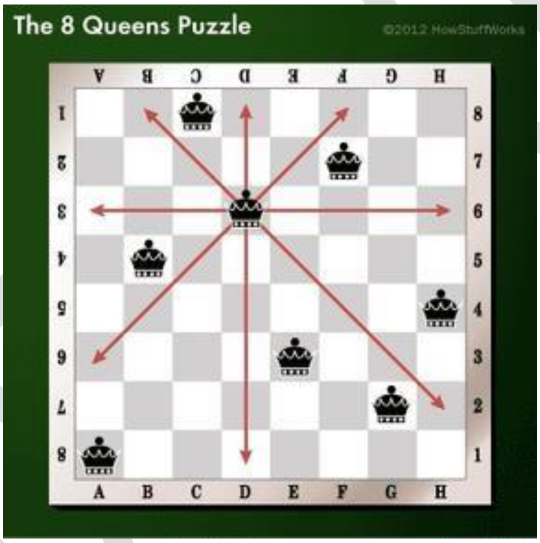
8- QUEENS PROBLEM

AIM:

To implement an 8-Queens problem using Python.

You are given an 8x8 board; find a way to place 8 queens such that no queen can attack any other queen on the chessboard. A queen can only be attacked if it lies on the same row, same column, or the same diagonal as any other queen. Print all the possible configurations.

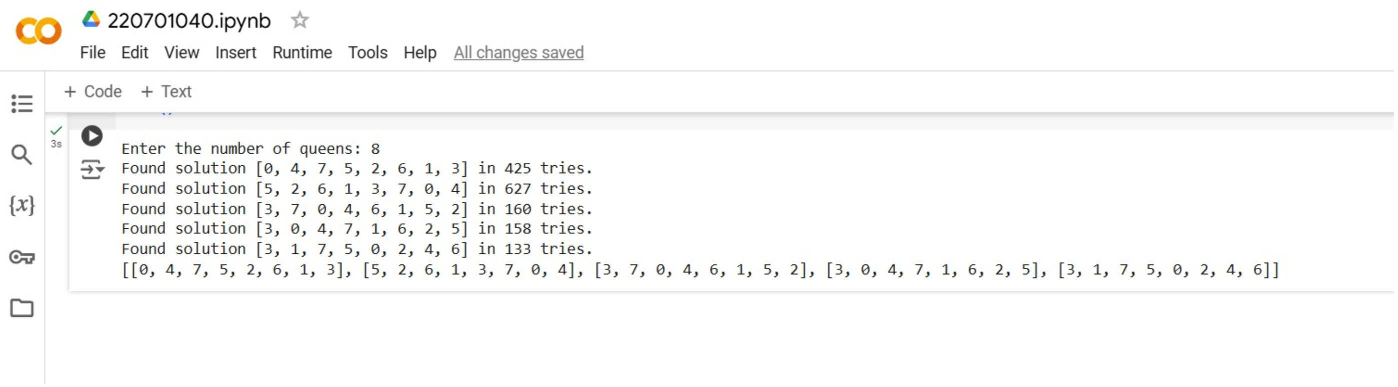
To solve this problem, we will make use of the Backtracking algorithm. The backtracking algorithm, in general checks all possible configurations and test whether the required result is obtained or not. For the given problem, we will explore all possible positions the queens can be relatively placed at. The solution will be correct when the number of placed queens = 8.



PROGRAM:

```
def share_diagonal(x0, y0, x1, y1):
   dx = abs(x0 - x1)
   dy = abs(y0 - y1)
   return dy == dx
def col_clashes(bs, c):
   for i in range(c):
      if share_diagonal(i, bs[i], c, bs[c]):
         return True
   return False
def has_clashes(the_board):
   for col in range(1, len(the_board)):
      if col_clashes(the_board, col):
         return True
   return False
def main():
   import random
   n=int(input("Enter the number of queens: "))
   rng = random.Random()
   bd = list(range(n))
   num_found = 0
   tries = 0
   result = []
   while num_found < 5:
      rng.shuffle(bd)
      tries += 1
      if not has_clashes(bd) and bd not in result:
         print("Found solution {0} in {1} tries.".format(bd, tries))
         tries = 0
         num_found += 1
         result.append(list(bd))
   print(result)
main()
```

OUTPUT:



```
Enter the number of queens: 8
Found solution [0, 4, 7, 5, 2, 6, 1, 3] in 425 tries.
Found solution [5, 2, 6, 1, 3, 7, 0, 4] in 627 tries.
Found solution [3, 7, 0, 4, 6, 1, 5, 2] in 160 tries.
Found solution [3, 0, 4, 7, 1, 6, 2, 5] in 158 tries.
Found solution [3, 1, 7, 5, 0, 2, 4, 6] in 133 tries.
[[0, 4, 7, 5, 2, 6, 1, 3], [5, 2, 6, 1, 3, 7, 0, 4], [3, 7, 0, 4, 6, 1, 5, 2], [3, 0, 4, 7, 1, 6, 2, 5], [3, 1, 7, 5, 0, 2, 4, 6]]
```

RESULT:

Thus, the 8Queens problem was implemented successfully using backtracking algorithm.