



Description

Editorial

Solutions (4.3K)

Submissions

Python3

Auto



Accepted



Runtime

Details

64 ms

Beats 83.01% of users with Python3

Memory

16.78 mb

Beats 82.07%

Next question

1. Two Sum

More challenges

52. N-Queens II

1001. Grid Illumination

All statuses

All languages

Time

Accepted

Python3

a few

Accepted

Python3

a few

Accepted

Python3

2 ms

Accepted

Python3

3 ms

Time Limit Exceeded

Python3

4 ms

Accepted

Python3

6 ms

Accepted

Python3

6 ms

Accepted

Python3

7 ms

Accepted

Python3

7 ms

Accepted

Python3

14

```
1 class Solution:
2     def solveNQueens(self, n: int) -> List[List[str]]:
3         self.n = n
4         self.result = []
5         self.columns = [False] * n
6         self.diagonals1 = [False] * (2 * n - 1)
7         self.diagonals2 = [False] * (2 * n - 1)
8         self.board = [['.' for _ in range(n)] for _ in range(n)]
9         self.backtrack(0)
10        return self.result
11
12    def backtrack(self, row):
13        if row == self.n:
14            self.result.append([''.join(self.board[r] for r in range(n))])
15            return
16
17        for col in range(self.n):
18            if not self.columns[col] and not self.diagonals1[row + col] and not self.diagonals2[row - col]:
19                self.board[row][col] = 'Q'
20                self.columns[col] = self.diagonals1[row + col] = self.diagonals2[row - col] = True
21                self.backtrack(row + 1)
22                self.board[row][col] = '.'
23                self.columns[col] = self.diagonals1[row + col] = self.diagonals2[row - col] = False
24
```

Console



Run

Submit