

N-Queens

[My Submissions \(/problems/n-queens/submissions/\)](/problems/n-queens/submissions/)

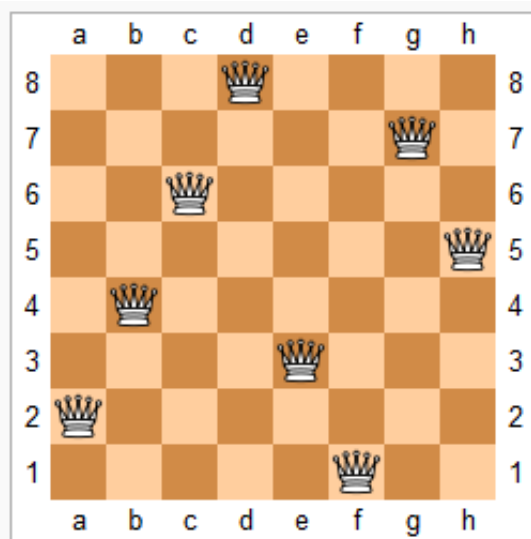
Total Accepted:

Question

Solution

39326 TotalSubmissions: **149182** Difficulty: **Hard**

The n -queens puzzle is the problem of placing n queens on an $n \times n$ chessboard such that no two queens attack each other.



One solution to the eight queens puzzle

Given an integer n , return all distinct solutions to the n -queens puzzle.

Each solution contains a distinct board configuration of the n -queens' placement, where 'Q' and '.' both indicate a queen and an empty space respectively.

For example,

There exist two distinct solutions to the 4-queens puzzle:

```
[
  [".Q..", // Solution 1
   "...Q",
   "Q...",
   "..Q."],

  [ "..Q.", // Solution 2
    "Q...",
    "...Q",
    ".Q.."]
]
```

[Show Tags](#)[Show Similar Problems](#)[✉ Send Feedback \(mailto:admin@leetcode.com?subject=Feedback\)](mailto:admin@leetcode.com?subject=Feedback)

Have you met this question in a real interview? [Discuss \(/discuss/questions/oj/n-queens/\)](/discuss/questions/oj/n-queens/)

Python ▼



```
1 class Solution(object):
2     def solveNQueens(self, n):
3         """
4         :type n: int
5         :rtype: List[List[str]]
6         """
7         returnColumn = []
8         column = [['.' for i in range(n)] for j in range(n)]
9         queens = [0 for i in range(n)]
10        self.DFS(column,0,n,queens,returnColumn)
11        return returnColumn
12    def DFS(self,column,step,n,queens,returnColumn):
13        if step == n:
14            newColumn = []
15            for i in range(n):
16                newColumn.append("".join(column[i]))
17            returnColumn.append(newColumn)
18            return
19        for i in range(n):
20            j=0
21            while j<step:
22                x = abs(step - j)
23                y = abs(i - queens[j])
24                if x==y:break
25                if x==0 or y==0:break
26                j+=1
27            if j>=step:
28                queens[step] = i
29                column[step][i] = 'Q'
30                self.DFS(column,step+1,n,queens,returnColumn)
31                column[step][i] = '.'
```

[Submit Solution](#)

Submission Result: Accepted (/submissions/detail/40199954/)

[More Details > \(/submissions/detail/40199954/\)](/submissions/detail/40199954/)Next challenges: [\(H\) N-Queens II \(/problems/n-queens-ii/\)](/problems/n-queens-ii/)

Share your acceptance!

[Frequently Asked Questions \(/faq/\)](#) | [Terms of Service \(/tos/\)](#)

[Privacy](#)

Copyright © 2015 LeetCode