

Rename notebook

+ <> + 

Connect



```
def solve_n_queens(n):  
    def is_safe(board, row, col):  
        for r in range(row):  
            c = board[r]  
            if c == col or abs(c - col) == abs(r - row):  
                return False  
        return True  
  
    def backtrack(row):  
        if row == n:
```



&gt; + T

Connect ▼



```
def backtrack(row):  
    if row == n:  
        result.append([(i + 1, board[i] + 1) for i in range(n)])  
        return  
    for col in range(n):  
        if is_safe(board, row, col):  
            board[row] = col  
            backtrack(row + 1)  
  
result = []  
board = [-1] * n
```





&gt; + T

Connect



```
result = []  
board = [-1] * n  
backtrack(0)  
return result
```

```
# Example usage
```

```
n = 4  
solutions = solve_n_queens(n)  
for solution in solutions:  
    print(solution)
```





Untitled98.ipynb



Rename notebook

< > +

Connect



```
solutions = solve_n_queens(n)
for solution in solutions:
    print(solution)
```



```
[(1, 2), (2, 4), (3, 1), (4, 3)]
[(1, 3), (2, 1), (3, 4), (4, 2)]
```