63010279 ณภัทร จิรารัตนกุลชัย

N-queens Problem (Recursive VS Iterative)

1) Source Code

1.1) Iterative

```
import time
class Board:
    def __init__(self, size):
        self.N = size
        self.queens = [] # list of columns, where the index represents the row

    def is_queen_safe(self, row, col):
        for r, c in enumerate(self.queens):
            if r == row or c == col or abs(row - r) == abs(col - c):
                return False
        return True

    def print_the_board(self):
        print ("solution:")
        for row in range(self.N):
            line = ['.'] * self.N
            if row < len(self.queens):
                line[self.queens[row]] = 'Q'
            print(".join(line))

    def solution(self):
        self.queens = []
        col = row = 0
        while True:
            while col < self.N and not self.is_queen_safe(row, col):
                col += 1
            if col < self.N:
                self.queens.append(col)
                if row + 1 >= self.N:
                    self.print_the_board()
                    self.queens.pop()
                    col = self.N
                else:
                    row += 1
                    col = 0
            if col >= self.N:
                # not possible to place a queen in this row anymore
                if row == 0:
                    return # all combinations were tried
                col = self.queens.pop() + 1
                row -= 1


if __name__ == "__main__":
    q = Board(5)
    q.solution()
```

Credit : algorithm - Avoid duplicates in N Queen iterative solutions (No Recursion Allowed) - Stack Overflow

1.2) Recursive

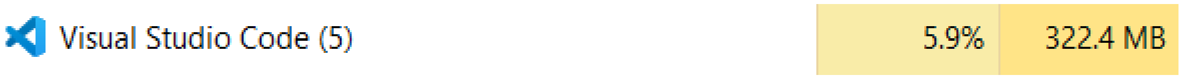
```
import time
def is_safe(board, x, y, c):
    for p in [board[i] for i in range(0, c)]:
        if p[0] == x or p[1] == y or x + y == p[0] + p[1] or x - y == p[0] - p[1]:
            return False
    return True

def nqueen_nrec(n):
    num = 0
    c = 0
    step = [0 for x in range(0, n + 1)]
    board = [(x, x) for x in range(0, n)]

    while c != -1:
        if c == n:
            num += 1
            print(board)
            c -= 1
            step[c] += 1
        elif step[c] == n:
            c -= 1
            step[c] += 1
        elif is_safe(board, step[c], c, c):
            board[c] = (step[c], c)
            c += 1
            step[c] = 0
        else:
            step[c] += 1
    print("Number of solution = {}".format(num))

if __name__ == "__main__":
    user_input = int(input("Enter input : "))
    t0 = time.time()
    c = nqueen_nrec(user_input)
    t1 = time.time()
    print(f'Time(Iteration): {t1-t0:.8} seconds')
```

Credit : algorithm - Avoid duplicates in N Queen iterative solutions (No Recursion Allowed) - Stack Overflow

2) CPU Memory

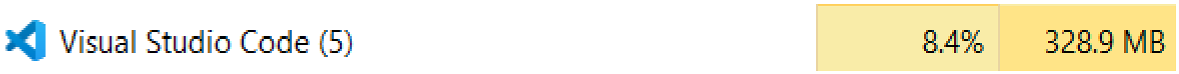
CPU Memory

2.1) Iterative



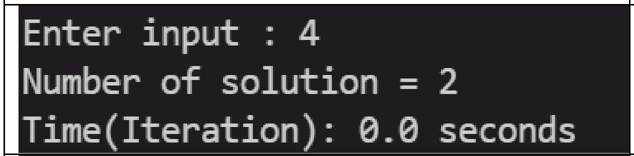
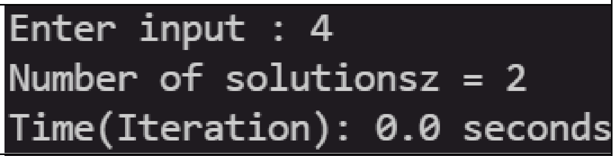
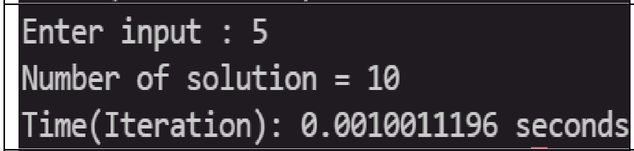
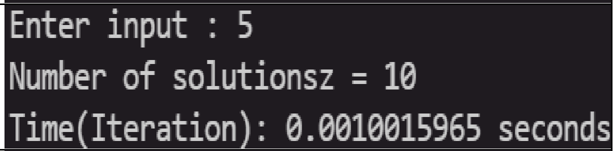
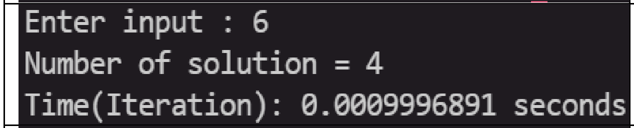
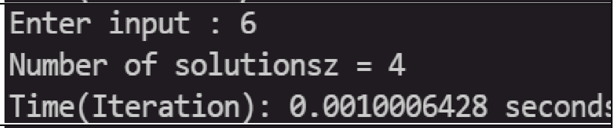
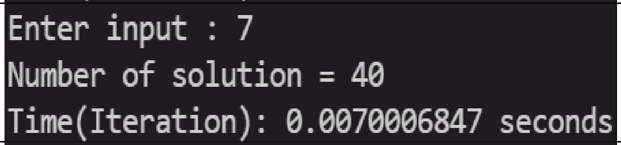
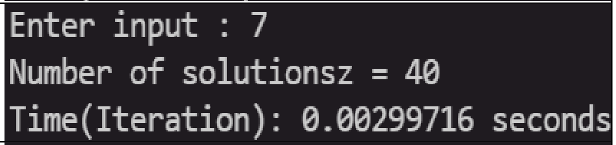
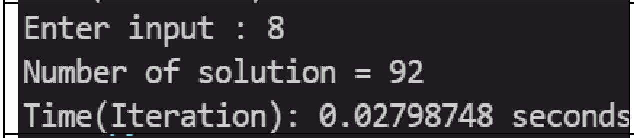
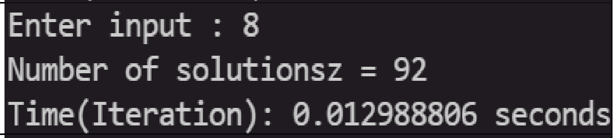
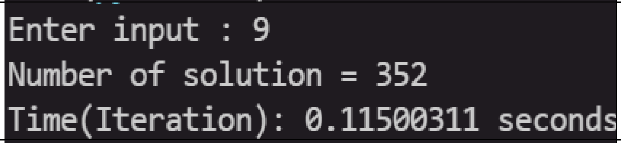
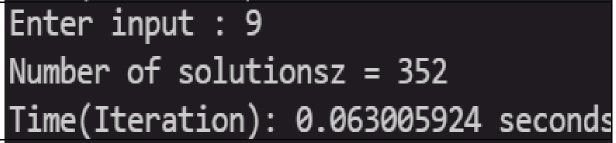
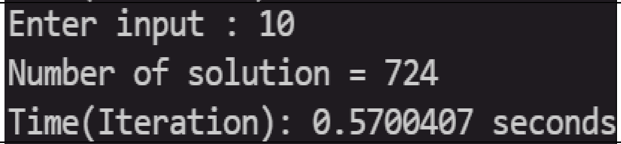
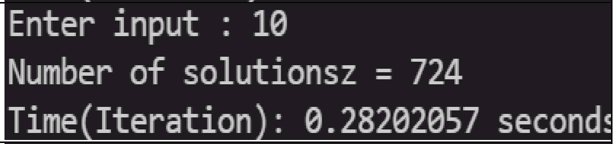
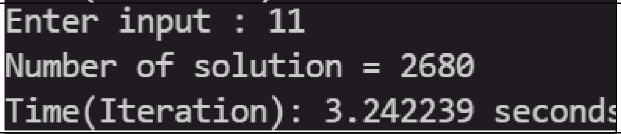
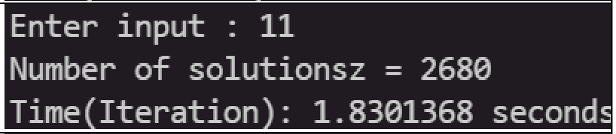
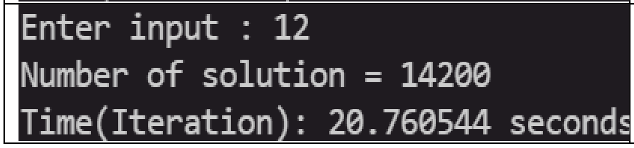
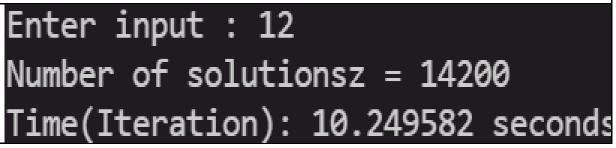
Visual Studio Code (5) 5.9% 322.4 MB

2.2) Recursive



Visual Studio Code (5) 8.4% 328.9 MB

3 & 4) ผลการรัน และการจับเวลา & Input 4 - 12

Iterative	Recursive
 Enter input : 4 Number of solution = 2 Time(Iteration): 0.0 seconds	 Enter input : 4 Number of solutionsz = 2 Time(Iteration): 0.0 seconds
 Enter input : 5 Number of solution = 10 Time(Iteration): 0.0010011196 seconds	 Enter input : 5 Number of solutionsz = 10 Time(Iteration): 0.0010015965 seconds
 Enter input : 6 Number of solution = 4 Time(Iteration): 0.0009996891 seconds	 Enter input : 6 Number of solutionsz = 4 Time(Iteration): 0.0010006428 seconds
 Enter input : 7 Number of solution = 40 Time(Iteration): 0.0070006847 seconds	 Enter input : 7 Number of solutionsz = 40 Time(Iteration): 0.00299716 seconds
 Enter input : 8 Number of solution = 92 Time(Iteration): 0.02798748 seconds	 Enter input : 8 Number of solutionsz = 92 Time(Iteration): 0.012988806 seconds
 Enter input : 9 Number of solution = 352 Time(Iteration): 0.11500311 seconds	 Enter input : 9 Number of solutionsz = 352 Time(Iteration): 0.063005924 seconds
 Enter input : 10 Number of solution = 724 Time(Iteration): 0.5700407 seconds	 Enter input : 10 Number of solutionsz = 724 Time(Iteration): 0.28202057 seconds
 Enter input : 11 Number of solution = 2680 Time(Iteration): 3.242239 seconds	 Enter input : 11 Number of solutionsz = 2680 Time(Iteration): 1.8301368 seconds
 Enter input : 12 Number of solution = 14200 Time(Iteration): 20.760544 seconds	 Enter input : 12 Number of solutionsz = 14200 Time(Iteration): 10.249582 seconds

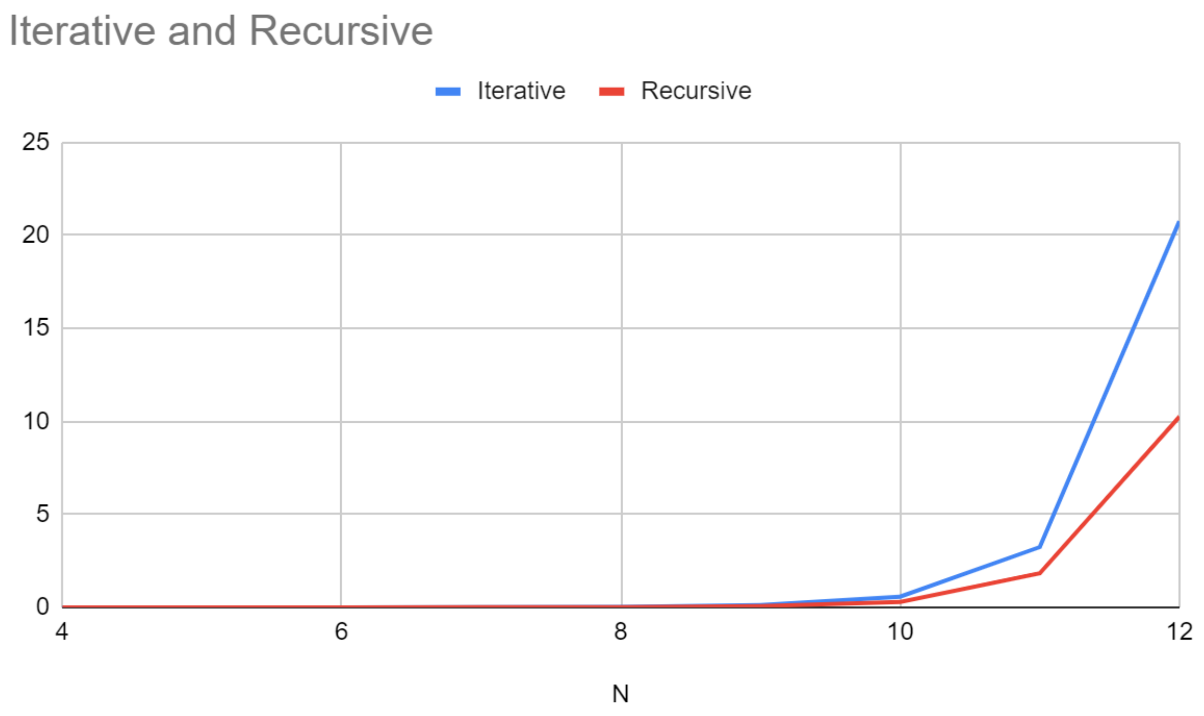
5) แหล่งอ้างอิง

Credit : algorithm - Avoid duplicates in N Queen iterative solutions (No Recursion Allowed) - Stack Overflow

6) ตารางบันทึกผล

N	Iterative	Recursive
4	0	0
5	0.0010011196	0.0010015965
6	0.0009996891	0.0010006428
7	0.0070006847	0.00299716
8	0.02798748	0.012988806
9	0.11500311	0.063005924
10	0.5700407	0.28202057
11	3.242239	1.8301368
12	20.760544	10.249582

7) กราฟเปรียบเทียบเวลาในการรัน ทั้ง สอง อัลกอริทึม



8) การวิเคราะห์ผลลัพธ์ที่ได้

สรุปผลได้ว่าแบบRecursiveจะใช้เวลาประมวลผลและใช้Memoryมากกว่าแบบIterative และRecursiveใช้ CPU มากกว่า(แต่ห่างกันไม่ค่อยมาก)