Iterative

```
[2, 1, 1, 1]
[1, 1, 1, 2]
[1, 2, 1, 1]
solution2

 Bord 4 x 4 --- 1.9927024841308594 milliseconds ---
```

```
 [2, 1, 1, 1, 1]
 [1, 1, 1, 2, 1]
 [1, 2, 1, 1, 1]
 solution10

  Bord 5 x 5 --- 7.978677749633789 milliseconds ---
```

```
 [1, 1, 1, 1, 1, 2]
 [1, 1, 1, 2, 1, 1]
 [1, 2, 1, 1, 1, 1]
 solution4

  Bord 6 x 6 --- 12.964963912963867 milliseconds ---
```

```
[1, 1, 1, 1, 1, 2, 1]
[1, 1, 1, 2, 1, 1, 1]
[1, 2, 1, 1, 1, 1, 1]
solution40

 Bord 7 x 7 --- 110.70394515991211 milliseconds ---
```

```
[1, 2, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 2, 1]
[1, 1, 1, 1, 2, 1, 1, 1]
solution92

  Bord 8 x 8 --- 1067.094326019287 milliseconds ---
```

```
[2, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 2, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 2, 1, 1, 1, 1]
solution352

 Bord 9 x 9 --- 8995.840549468994 milliseconds ---
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 2, 1]
[1, 1, 1, 1, 1, 1, 2, 1, 1, 1]
[1, 1, 1, 2, 1, 1, 1, 1, 1, 1]
solution724

 Bord 10 x 10 --- 103998.08621406555 milliseconds ---
```

Recursive

```
[1, 3, 0, 2]
[2, 0, 3, 1]
number of solutions =  2
 Bord 4 x 4 --- 0.8742809295654297 milliseconds ---
```

```
[3, 0, 2, 4, 1]
[3, 1, 4, 2, 0]
[4, 1, 3, 0, 2]
[4, 2, 0, 3, 1]
number of solutions =  10
 Bord 5 x 5 --- 0.9982585906982422 milliseconds ---
```

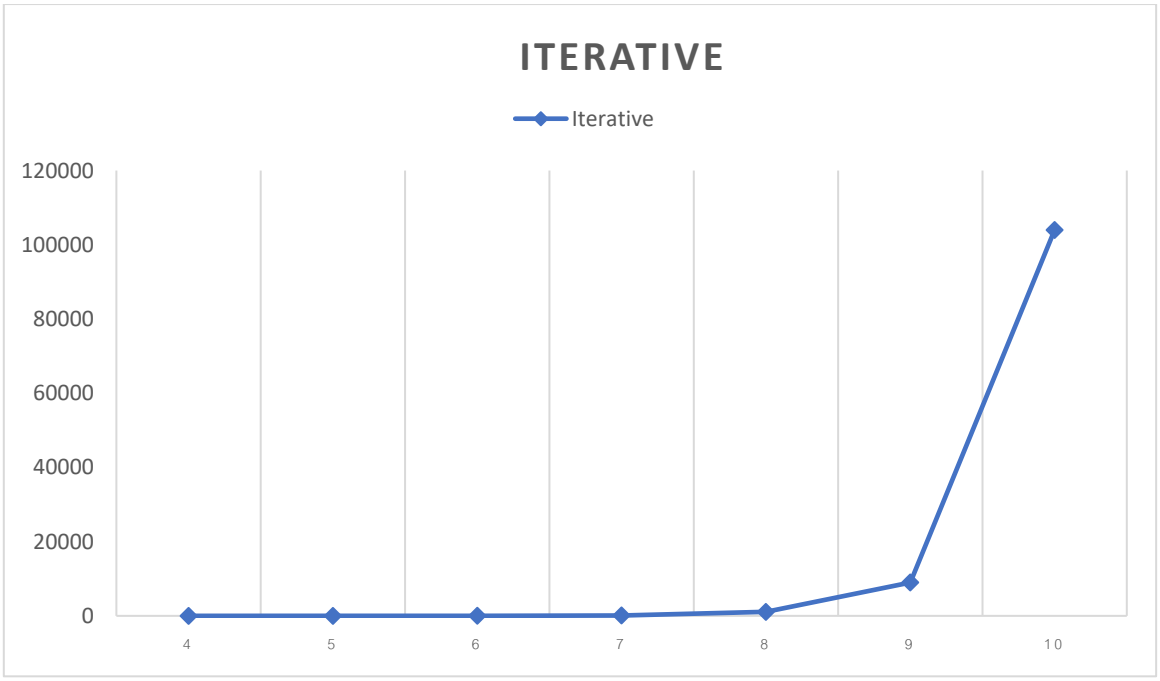
```
[1, 3, 5, 0, 2, 4]
[2, 5, 1, 4, 0, 3]
[3, 0, 4, 1, 5, 2]
[4, 2, 0, 5, 3, 1]
number of solutions =  4
 Bord 6 x 6 --- 2.0029544830322266 milliseconds ---
```

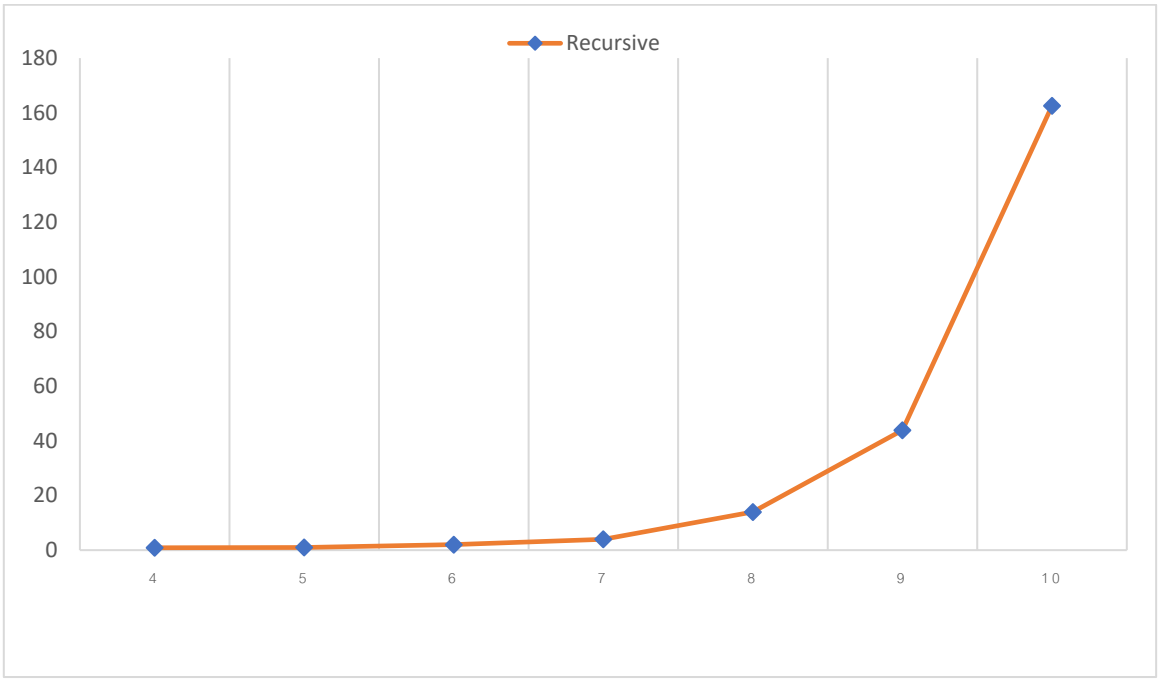
```
[6, 1, 3, 5, 0, 2, 4]
[6, 2, 5, 1, 4, 0, 3]
[6, 3, 0, 4, 1, 5, 2]
[6, 4, 2, 0, 5, 3, 1]
number of solutions =  40
 Bord 7 x 7 --- 3.9894580841064453 milliseconds ---
```

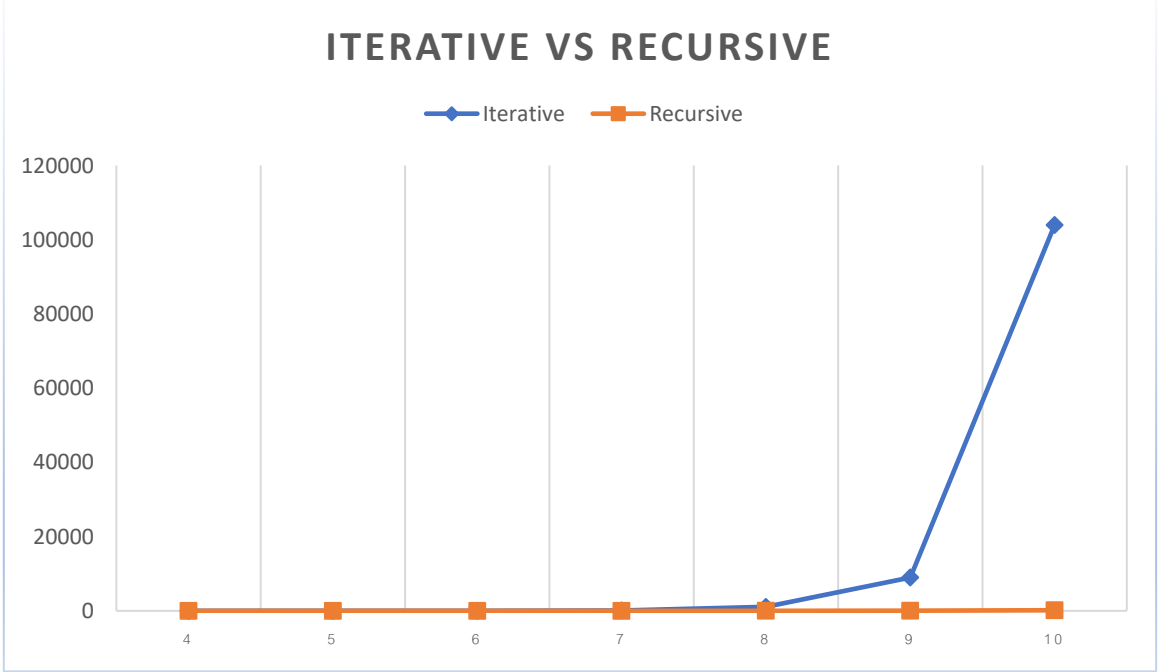
```
[7, 1, 3, 0, 6, 4, 2, 5]
[7, 1, 4, 2, 0, 6, 3, 5]
[7, 2, 0, 5, 1, 4, 6, 3]
[7, 3, 0, 2, 5, 1, 6, 4]
number of solutions =  92
 Bord 8 x 8 --- 13.964414596557617 milliseconds ---
```

```
[8, 5, 7, 1, 3, 0, 6, 4, 2]
[8, 6, 1, 3, 0, 7, 4, 2, 5]
[8, 6, 2, 7, 1, 4, 0, 5, 3]
[8, 6, 3, 1, 7, 5, 0, 2, 4]
number of solutions =  352
 Bord 9 x 9 --- 43.883323669433594 milliseconds ---
```

```
[9, 7, 1, 3, 0, 6, 8, 5, 2, 4]
[9, 7, 4, 1, 3, 0, 6, 8, 2, 5]
[9, 7, 4, 1, 3, 0, 6, 8, 5, 2]
[9, 7, 4, 2, 0, 5, 1, 8, 6, 3]
number of solutions =  724
 Bord 10 x 10 --- 162.5652313232422 milliseconds ---
```


ITERATIVE
Iterative
120000
100000
80000
60000
40000
20000
0
4
5
6
7
8
9
10


Recursive
180
160
140
120
100
80
60
40
20
0
4
5
6
7
8
9
10



CPU Memory : intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz

แหล่งอ้างอิง : https://colab.research.google.com/drive/1nhVvTij1LuF-nB1okf9MHtyTdpmARzdG

วิเคราะห์ผลลัพธ์ : การเขียนแบบ Recursive จะใช้Runtime น้อยกว่าแบบ Iterative แต่ว่าในช่วงแรกจะมีระยะเวลาที่ใกล้เคียงกัน แต่ว่าเมื่อ Nมากขึ้น iterativeจะช้ากว่าเพราะจะทำทุกความเป็นไปได้แต่Recursiveจะทำงานฉพาะกรณีที่เป็นไปได้ เวลาที่ทำงานจึงลดลง