N QUEEN

4 Queen (ready)

x

y

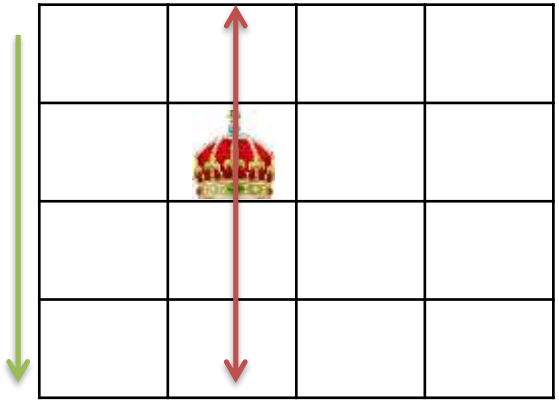
1.1	1.2	1.3	1.4
2.1	2.2	2.3	2.4
3.1	3.2	3.3	3.4
4.1	4.2	4.3	4.4

- Each Y line has one queen
- Each X column has one queen

- Start working by placing one queen in row Y
- **For example,**
- There is a queen on 2,2
  ① A queen cannot be placed in the same row x.

x

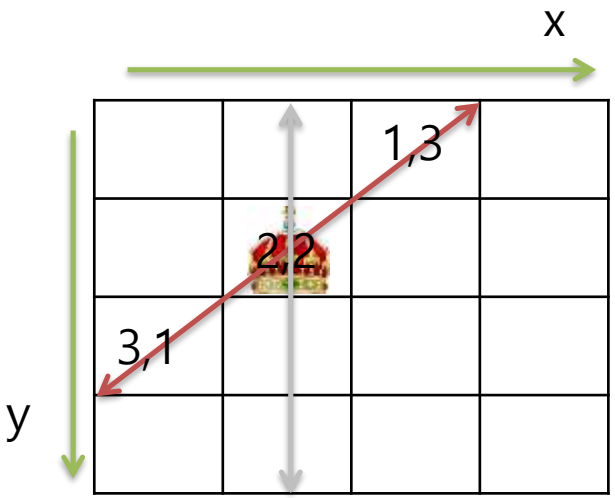
y



visitedX

1	2	3	4
	1		

4 Queen (ready)

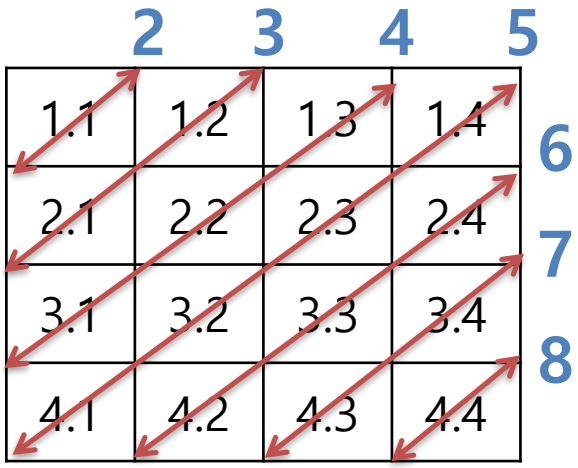


- There is a queen on 2,2
  - ① A queen cannot be placed in the same row x.
  - ② It cannot be on the same upward diagonal. In the case of (2,2) it shall not be located in (1,3) (3,1).
    - **All of these have an equal value of 4 for row plus row (y+x).**

visitedIncrease

1	2	3	4	5	6	7	8
			1				

4 Queen (ready)

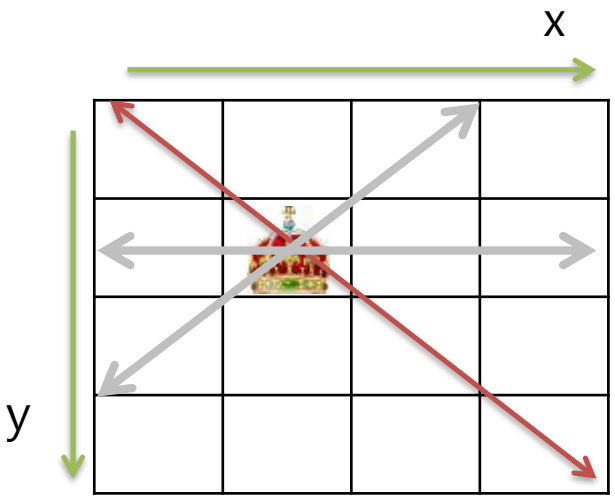


- visitedIncrease[y+x]

visitedIncrease

1	2	3	4	5	6	7	8
			1				

4 Queen (ready)



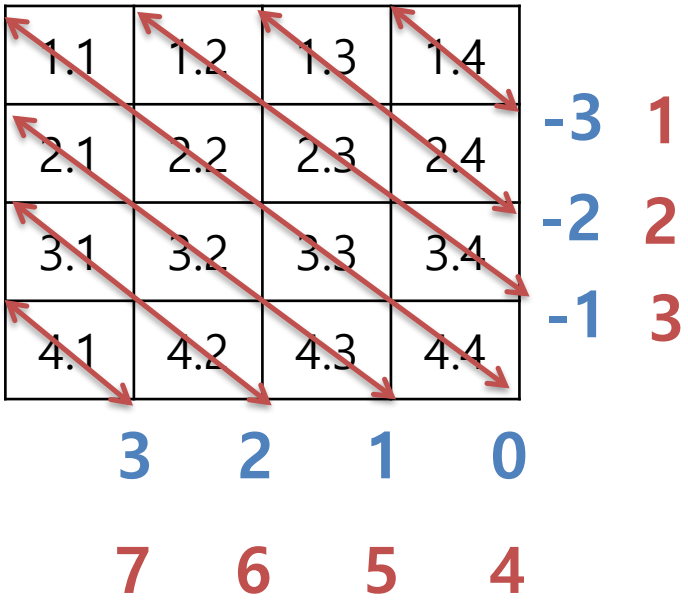
- There is a queen on 2,2
  - ① A queen cannot be placed in the same row x.
  - ② It cannot be on the same upward diagonal.
  - ③ It cannot be on the same downward diagonal.

1.1	1.2	1.3	1.4
2.1	2.2	2.3	2.4
3.1	3.2	3.3	3.4
4.1	4.2	4.3	4.4

visitedDecrease

1	2	3	4	5	6	7	8

4 Queen (ready)

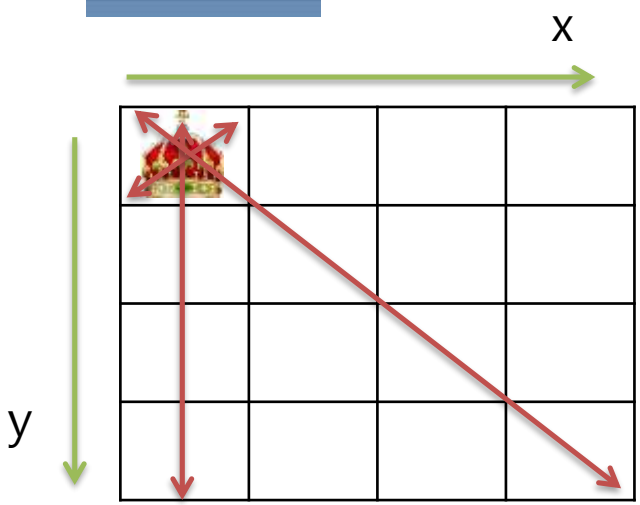


- Add 4 as the value of '-' is cumbersome.
- y - x + 4
- visitedDecrease[y-x]

visitedDecrease

1	2	3	4	5	6	7	8
			1				

4 Queen (y = 1)



y = 1

x = 1

y + x = 1 + 1 = 2
visitedIncrease[2] = 1

y – x + 4 = 4
visitedIncrease[4] = 1

visitedX

1	2	3	4
1			

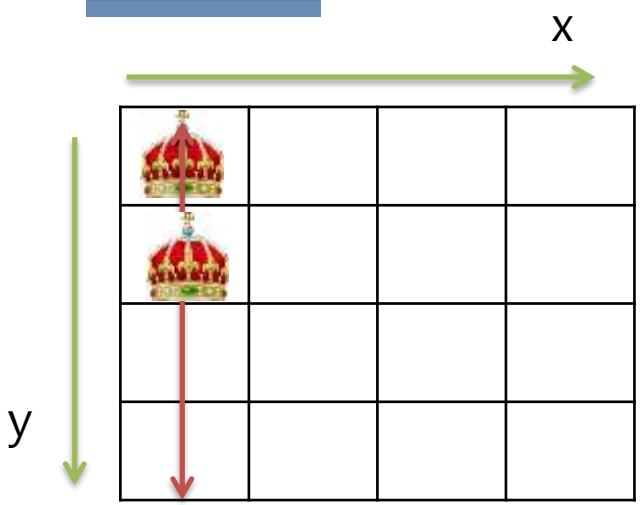
visitedIncrease

1	2	3	4	5	6	7	8
	1						

visitedDecrease

1	2	3	4	5	6	7	8
			1				

4 Queen (y = 2)



y = 2

X = 1

visitedX

1	2	3	4
1			

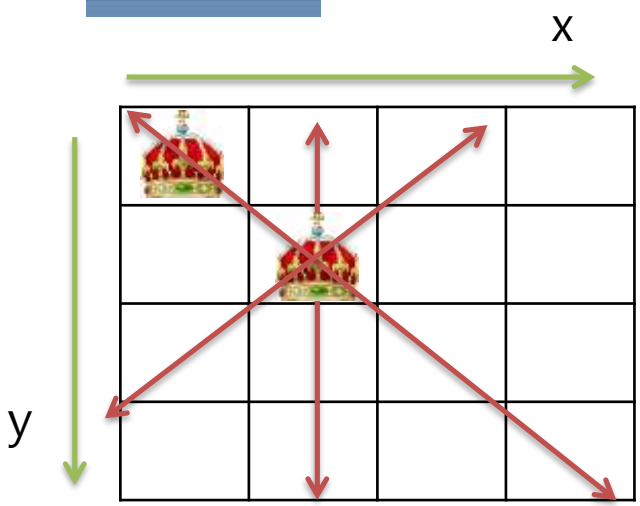
visitedIncrease

1	2	3	4	5	6	7	8
	1						

visitedDecrease

1	2	3	4	5	6	7	8
			1				

4 Queen (y = 2)



y = 2

X = 2

y + x = 2 + 2 = 4
visitedIncrease[4] = 1

y – x + 4 = 2 -2 +4 = 4
visitedIncrease[4] = 1

visitedX

1	2	3	4
1	1		

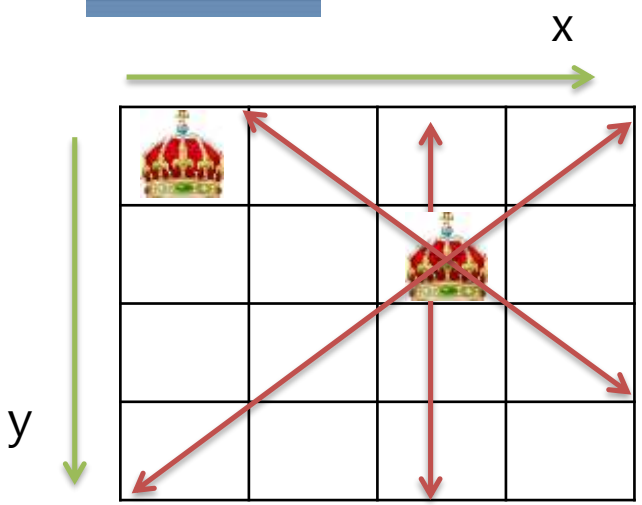
visitedIncrease

1	2	3	4	5	6	7	8
	1		1				

visitedDecrease

1	2	3	4	5	6	7	8
			1				

4 Queen (y = 2)



y = 2

X = 3

y + x = 2 + 3 = 5
visitedIncrease[5] = 1

y – x + 4 = 2 – 3 + 4 = 3
visitedIncrease[3] = 1

visitedX

1	2	3	4
1		1	

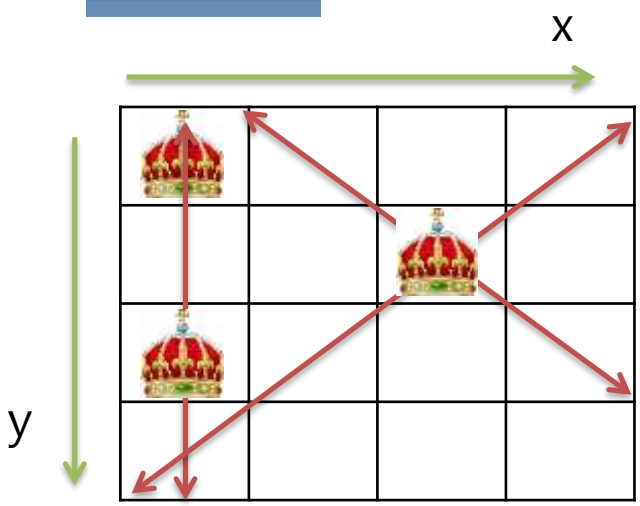
visitedIncrease

1	2	3	4	5	6	7	8
	1			1			

visitedDecrease

1	2	3	4	5	6	7	8
		1	1				

4 Queen (y = 3)



y = 3

X = 1

visitedX

1	2	3	4
1		1	

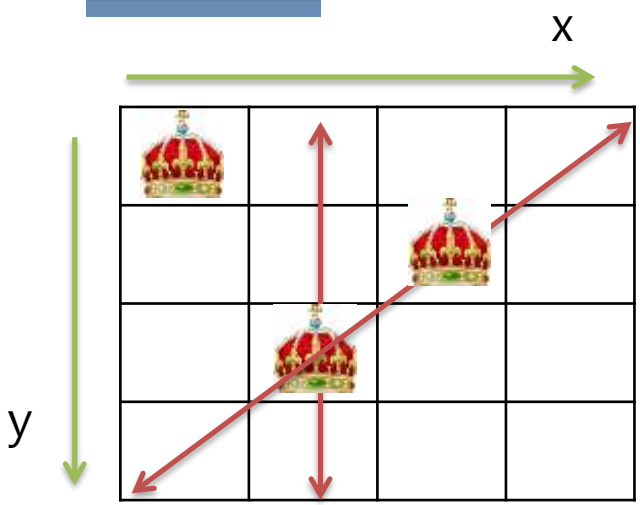
visitedIncrease

1	2	3	4	5	6	7	8
	1			1			

visitedDecrease

1	2	3	4	5	6	7	8
		1	1				

4 Queen (y = 3)



y = 3

X = 2

y + x = 3 + 2 = 5
visitedIncrease[5] = 1

visitedX

1	2	3	4
1	1	1	

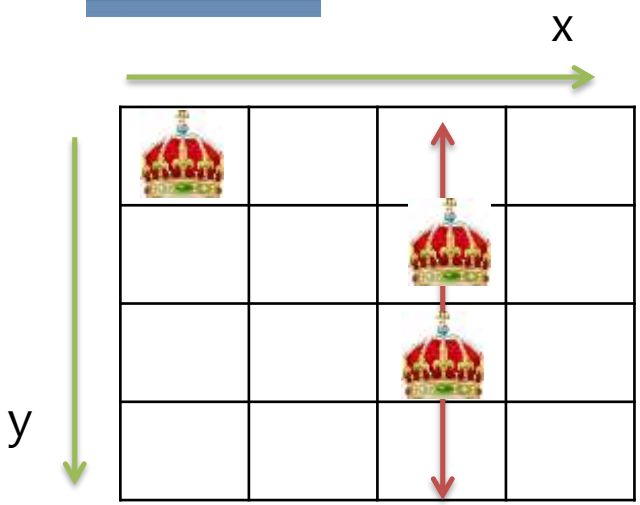
visitedIncrease

1	2	3	4	5	6	7	8
	1			1			

visitedDecrease

1	2	3	4	5	6	7	8
		1	1				

4 Queen (y = 3)



y = 3

x = 3

visitedX

1	2	3	4
1		1	

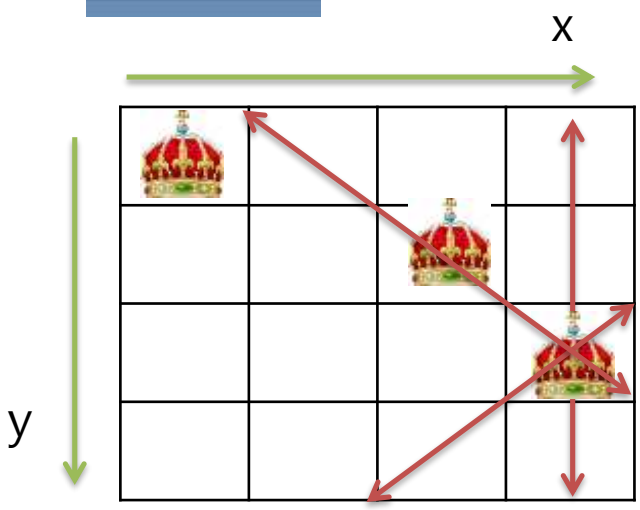
visitedIncrease

1	2	3	4	5	6	7	8
	1			1			

visitedDecrease

1	2	3	4	5	6	7	8
		1	1				

4 Queen (y = 3)



y = 3

x = 4

y + x = 3 + 4 = 7
visitedIncrease[7] = 1

y – x + 4 = 3 – 4 + 4 = 3
visitedIncrease[3] = 1

visitedX

1	2	3	4
1		1	1

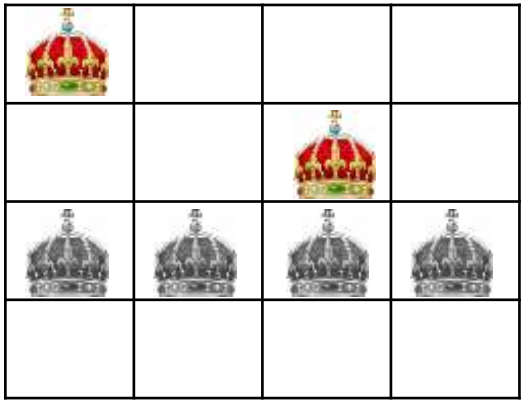
visitedIncrease

1	2	3	4	5	6	7	8
	1			1		1	

visitedDecrease

1	2	3	4	5	6	7	8
		1	1				

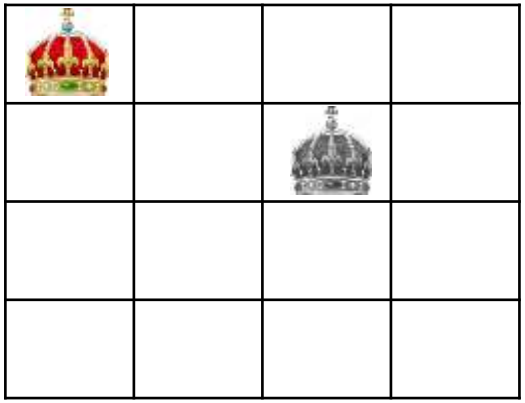
4 Queen (Backtracking)



- You cannot place a queen in every column of the third row.
- Go back to the second row and try the column you didn't try.

- All information about the queen who was placed in column 3 of the second row must be deleted.

4 Queen (Backtracking)



- All information about the queen who was placed in column 2 of the second row must be deleted.

visitedX

1	2	3	4
1		1	

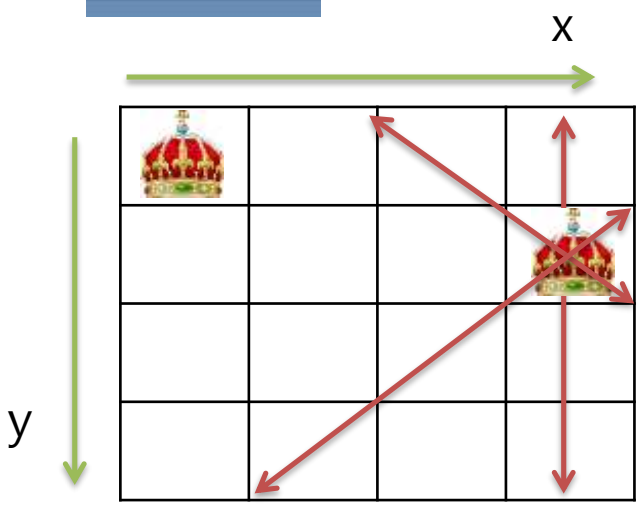
visitedIncrease

1	2	3	4	5	6	7	8
	1			1			

visitedDecrease

1	2	3	4	5	6	7	8
		1	1				

4 Queen (y = 2)



y = 2

x = 4

y + x = 2 + 4 = 6
visitedIncrease[6] = 1

y – x + 4 = 2 – 4 + 4 = 2
visitedIncrease[2] = 1

visitedX

1	2	3	4
1			1

visitedIncrease

1	2	3	4	5	6	7	8
	1				1		

visitedDecrease

1	2	3	4	5	6	7	8
	1		1				

nQueen

```
ans = 0
visitedX = [0]*10
visitedIncrease = [0]*10
visitedDecrease = [0]*10

def GetSome(y):
    global ans
    if y > 4:
        ans +=1
        return

    for x in range(1,5):
        if not visitedX[x] and not visitedIncrease[y+x] and not visitedDecrease[y-x+4]:
            visitedX[x] = visitedIncrease[y+x] = visitedDecrease[y-x+4] = True
            GetSome(y+1)
            visitedX[x] = visitedIncrease[y + x] = visitedDecrease[y - x + 4] = False


GetSome(1)
print(ans)
```