



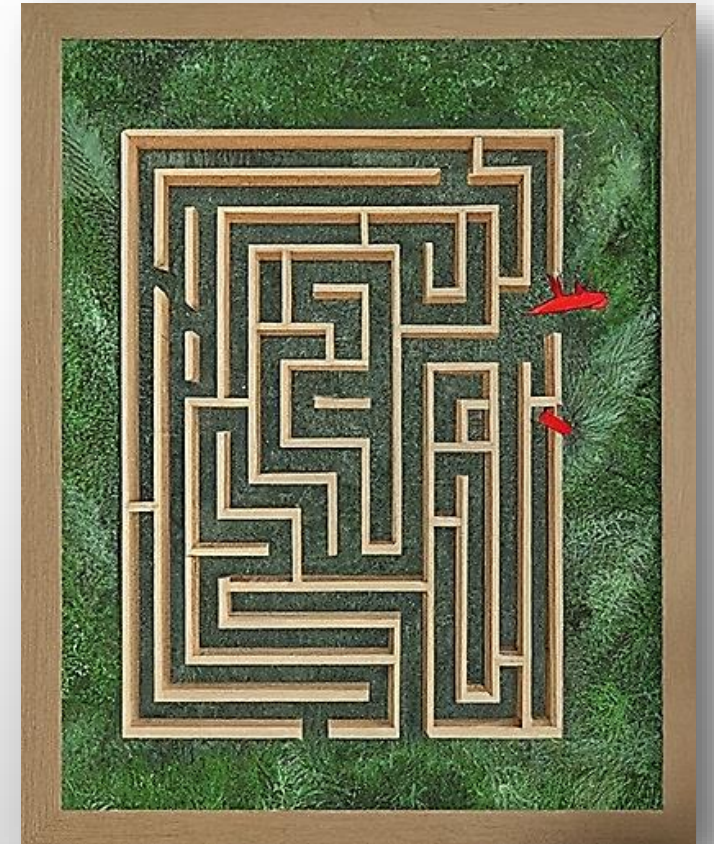
# **Bölüm 13: Geri Dönüşlü Algoritmalar**

## **Algoritmalar**



# Geri Dönüp Dene

- Bazen bir problemi çözmek için
  - farklı yollar denemek ve
  - yanlış yollardan geri dönmek gerekir.
- Backtracking, karmaşık problemleri çözmek için bu stratejiyi kullanır.





# Backtracking Algoritmaları

- Bir labirentte yolculuk yapan kişiye benzetilebilir.
- Kişi, her kavşak noktasında bir karar vererek ilerler.
- Seçtiği yol çıkışa götürmüyorsa, geri dönüp başka bir yol denenir.
- Her adımda bir karar verilir ve kararın doğru olup olmadığı kontrol edilir.
- Bu deneme-yanılma süreci problemi çözene kadar sürer.
- Labirent, sudoku ve satranç gibi yapay zeka oyunlarında kullanılır.



# Sudoku

- Sudoku'nun amacı,
  - 9x9'luk bir kare içindeki her satır, her sütun ve her 3x3'lük küçük karede
  - 1'den 9'a kadar olan sayıları tek seferde kullanmaktır.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku

- Boş bir hücreye bir rakam ata.
- Eğer atanan rakam geçerli ise, devam et.
- Geçerli değilse, geri adım at (backtrack).
- Önceki adıma dön ve farklı bir rakam dene.
- Eğer tüm rakamlar denenmişse,
  - önceki hücreye geri dön
  - farklı bir rakam dene.
- Tüm hücreler dolana kadar adımları tekrarla.

# Sudoku



5	3	1	2	7	6	8	9	4
6	2	4	1	9	5	2		
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku



5	3	1	2	7	6	9	4	8
6	4	2	1	9	5	7	3	4
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	1	6	7	2	4	9	8
6	4	7	1	9	5	3		
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku



5	3	1	6	7	2	8	9	4
6	7	4	1	9	5	6		
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	1	6	7	4	8	9	2
6	2	4	1	9	5	7	7	
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	1	6	7	8	9	2	4
6	2	4	1	9	5	8	6	
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	1	6	7	8	9	4	2
6	2	4	1	9	5	7	3	8
8	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	2		7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku



5	3	2	6	7	8	1	9	4
6	4	7	1	9	5	3	2	8
1	9	8	2	2			6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku



5	3	2	6	7	8	9	1	4
6	7	4	1	9	5	8	3	2
1	9	8	3	4	9		6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	4	2	7	6	9	1	8
6	2	7	1	9	5	4	4	
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	4	6	7	2	8	9	3
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	4	6	7	8	1	9	2
6	2	7	1	9	5	3	4	8
1	9	8	2	3	4	5	6	7
8	1	2	9	6	7	4	5	3
4	5	6	8	1	3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku



5	3	4	6	7	8	9	1	2
6	2	7	1	9	5	4	3	8
1	9	8	2	3	4	5	6	7
8	1	9		6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	2	3	4	5	6	7
8	1	5	9	6	1			3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



# Sudoku



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	2	9	7	6	1	4	5	3
4	6		8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	3	2				6
	6					2	8	
			4	1	9			5
				8			7	9

# Sudoku



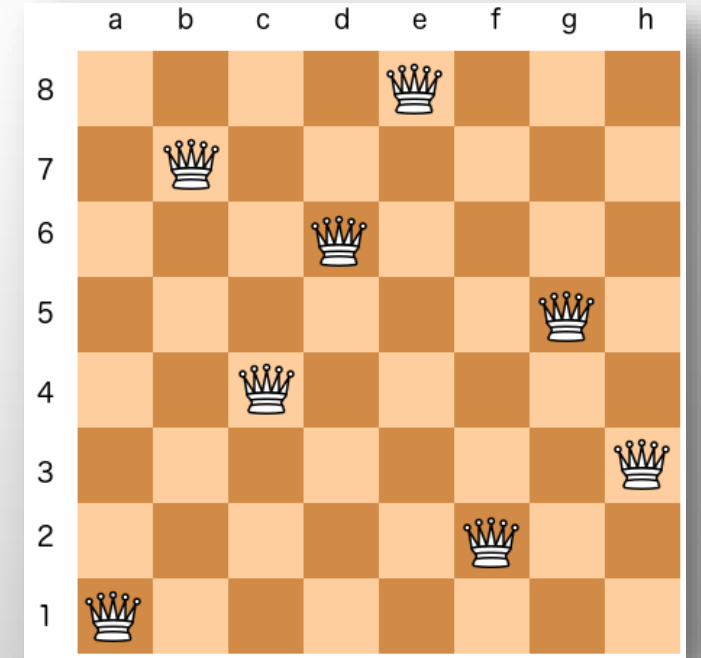
5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9





# 8 Vezir Problemi

- Bir 8x8 satranç tahtasına 8 vezirin yerleştirilmesi.
- Bulmaca, her bir vezirin diğerlerini tehdit etmediği bir yerleşim bulma.
- Backtracking algoritması, bu tür problemleri çözmek için kullanılır.





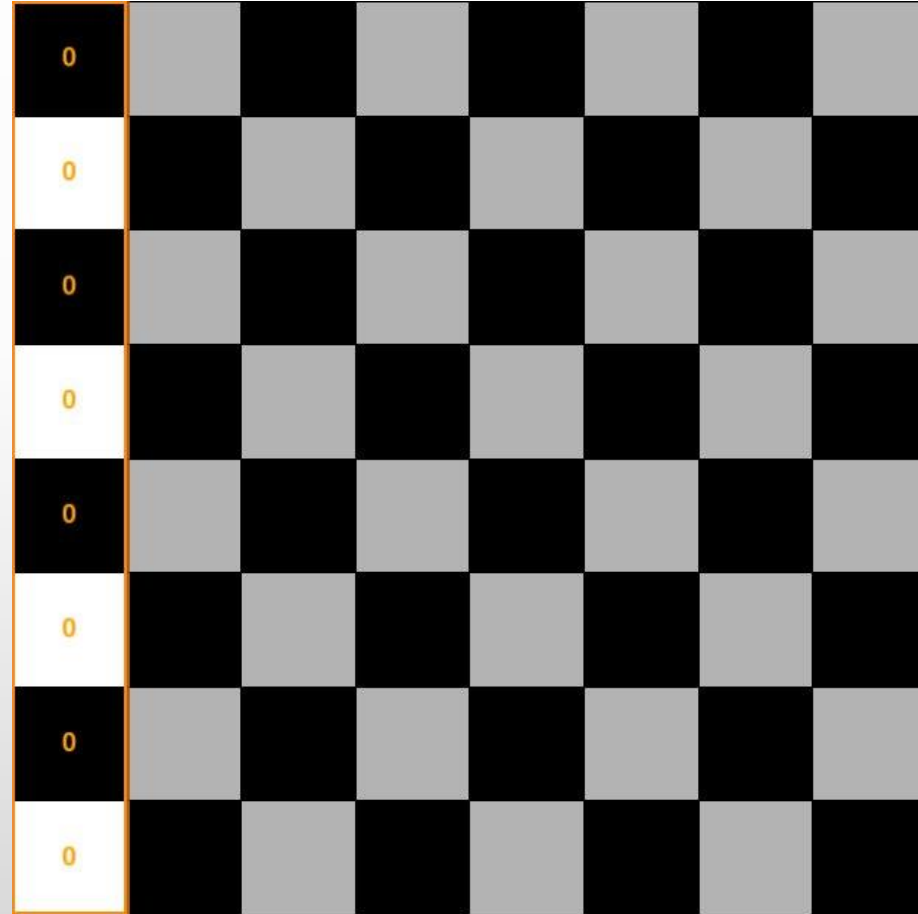


## 8 Vezir Problemi

- Tahtanın her hücresi başlangıçta boş olarak atanır.
- İlk vezir ilk sıradaki bir sütuna yerleştirilir.
- Yerleştirilen vezirin diğer vezirlerle çakışıp çakışmadığı kontrol edilir.
- Çakışma yoksa, bu konum geçerli kabul edilir.
- Bir sonraki vezir sonraki sırada bir sütuna yerleştirilir.
- Tüm sütunlar denenip çözüm bulunamazsa, bir önceki vezir yer değiştirilir.
- Tüm 8 vezir yerleştirildiğinde, çözüm bulunmuş olur.




# 8 Queens



# 8 Queens



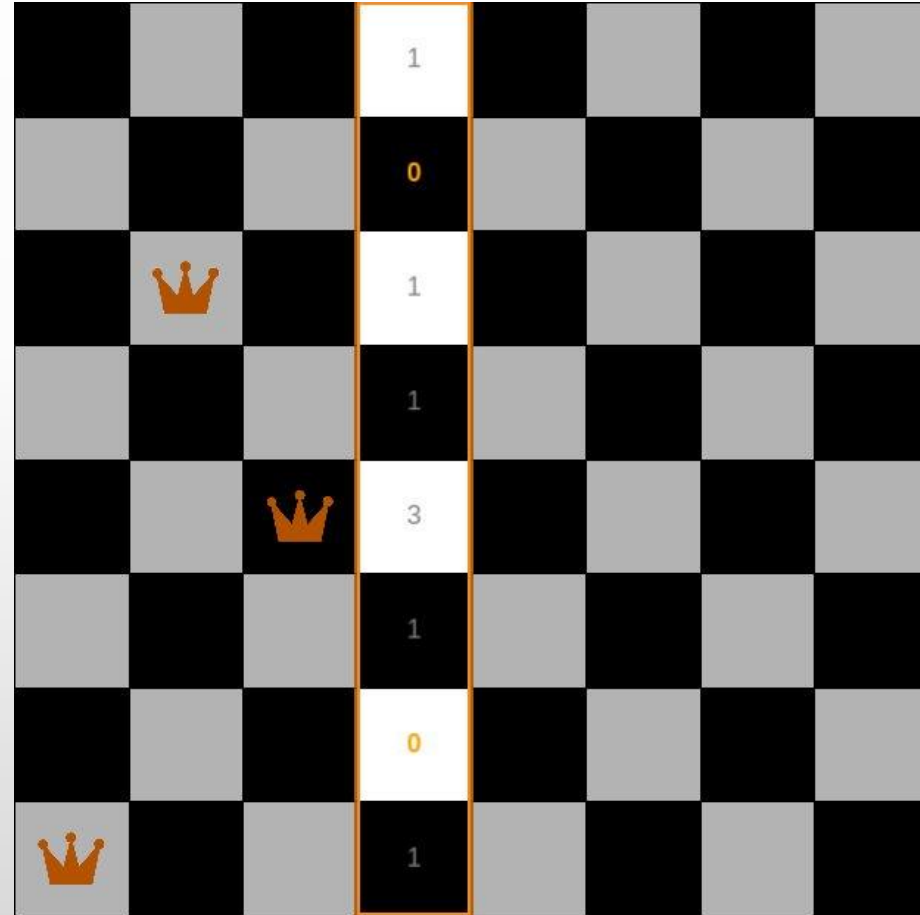
	0						
	0						
	0						
	0						
	0						
	0						
	1						
	1						

# 8 Queens

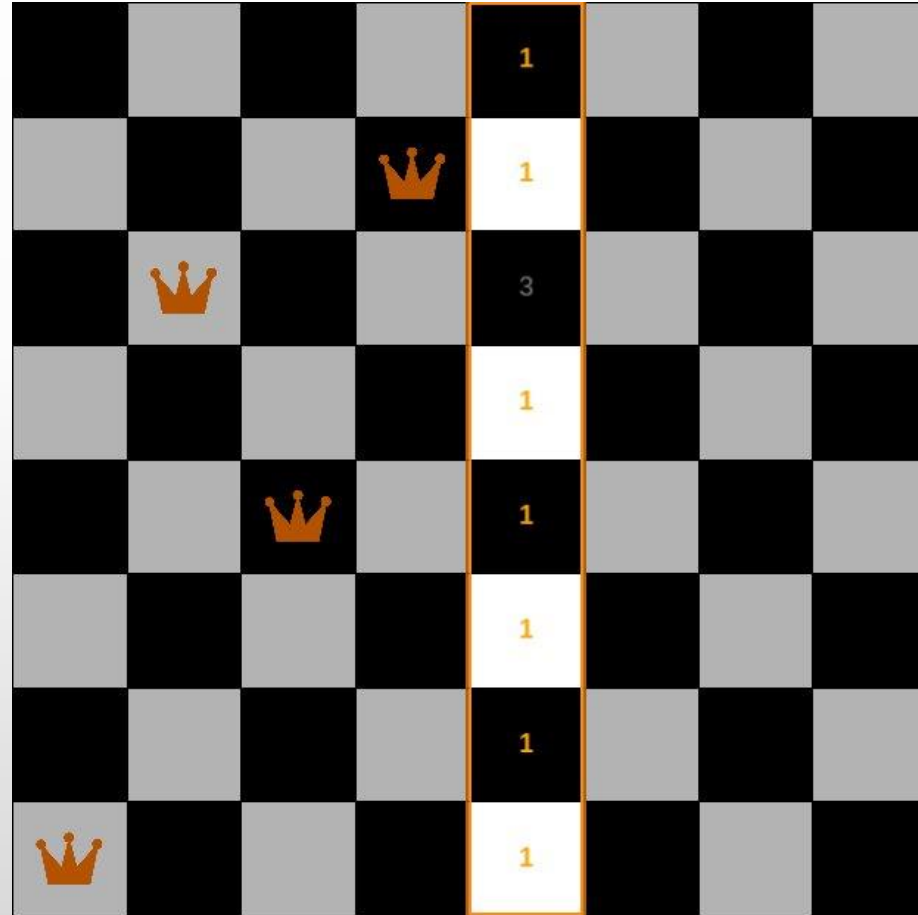


		0					
		1					
	♔	1					
		1					
		0					
		1					
		0					
♔		1					

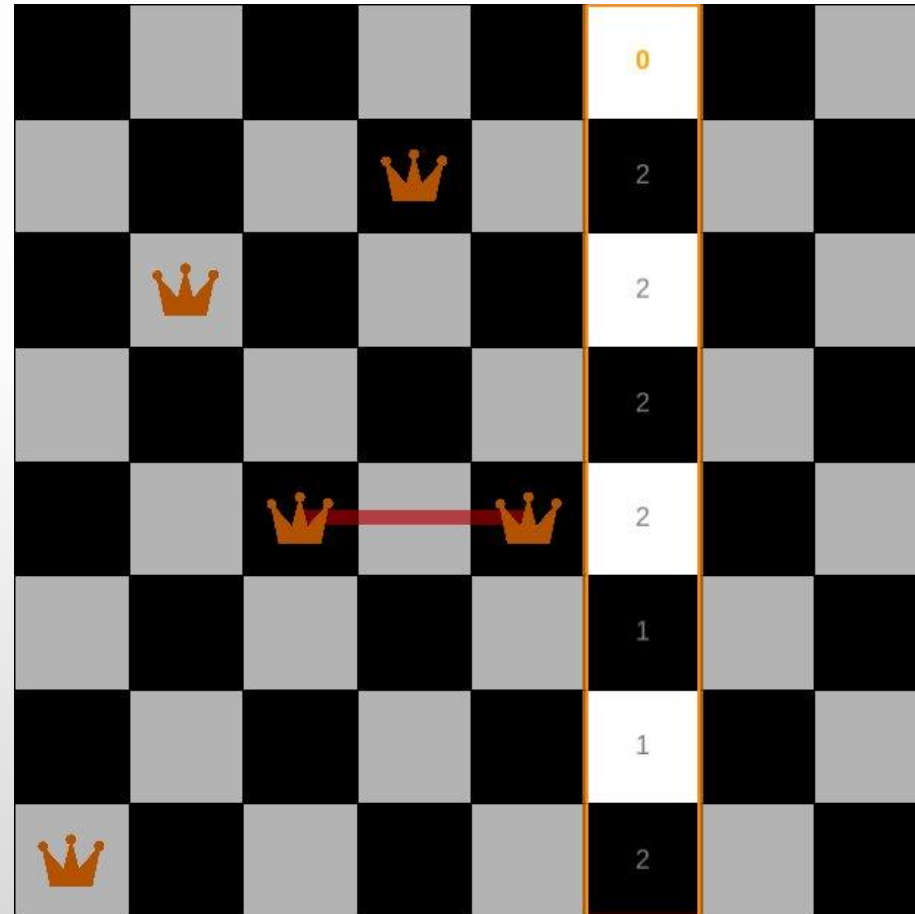
# 8 Queens



# 8 Queens

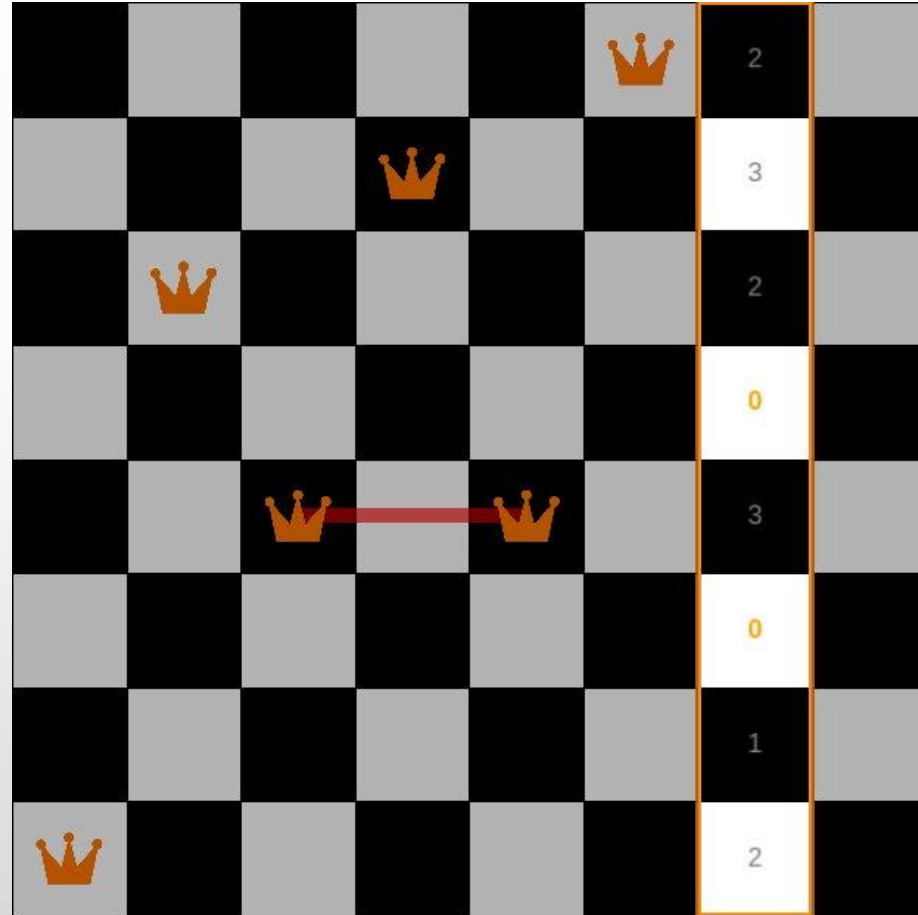


# 8 Queens

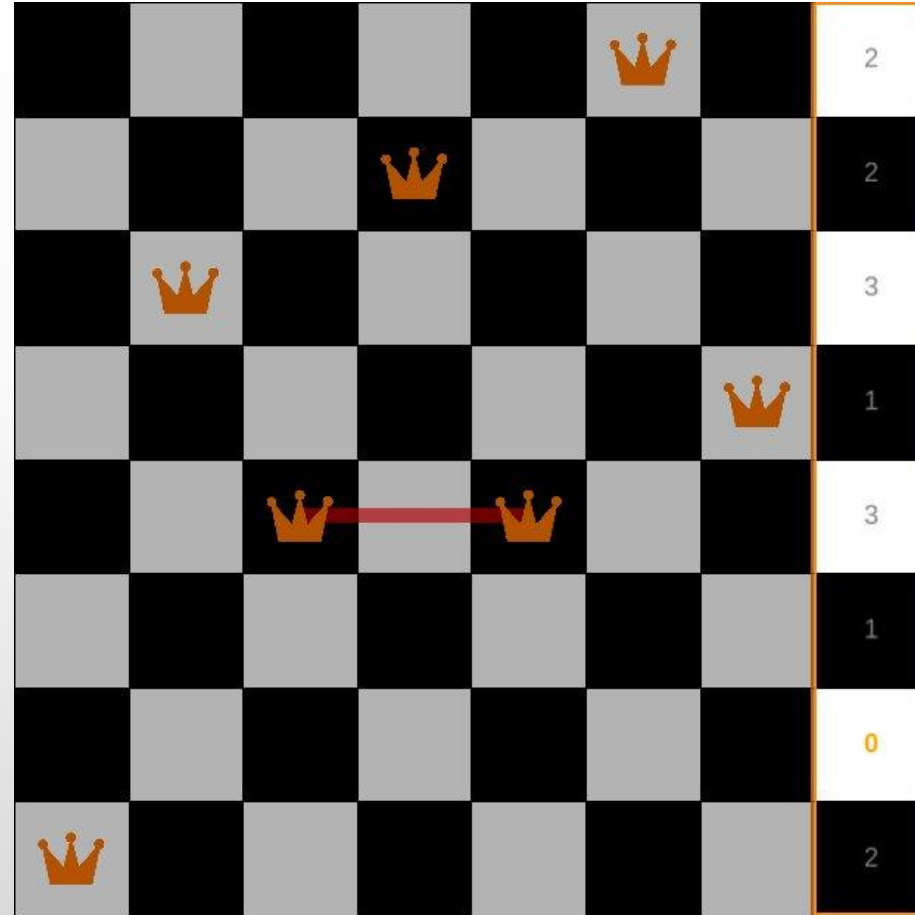




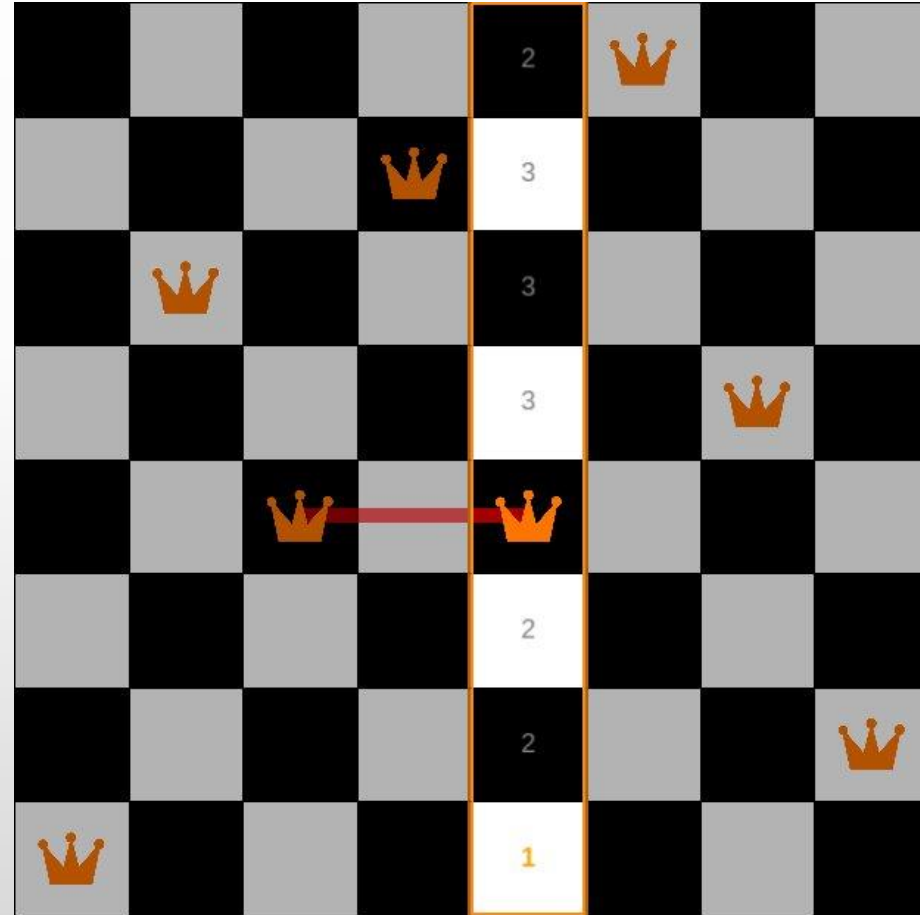
# 8 Queens



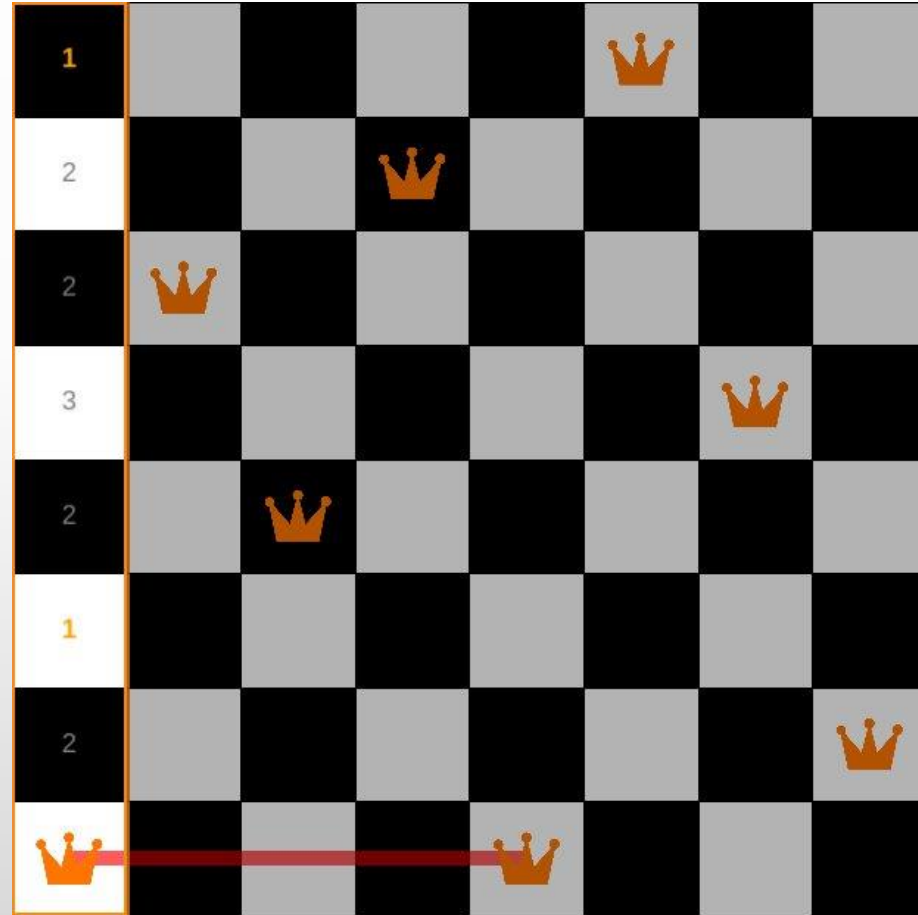
# 8 Queens



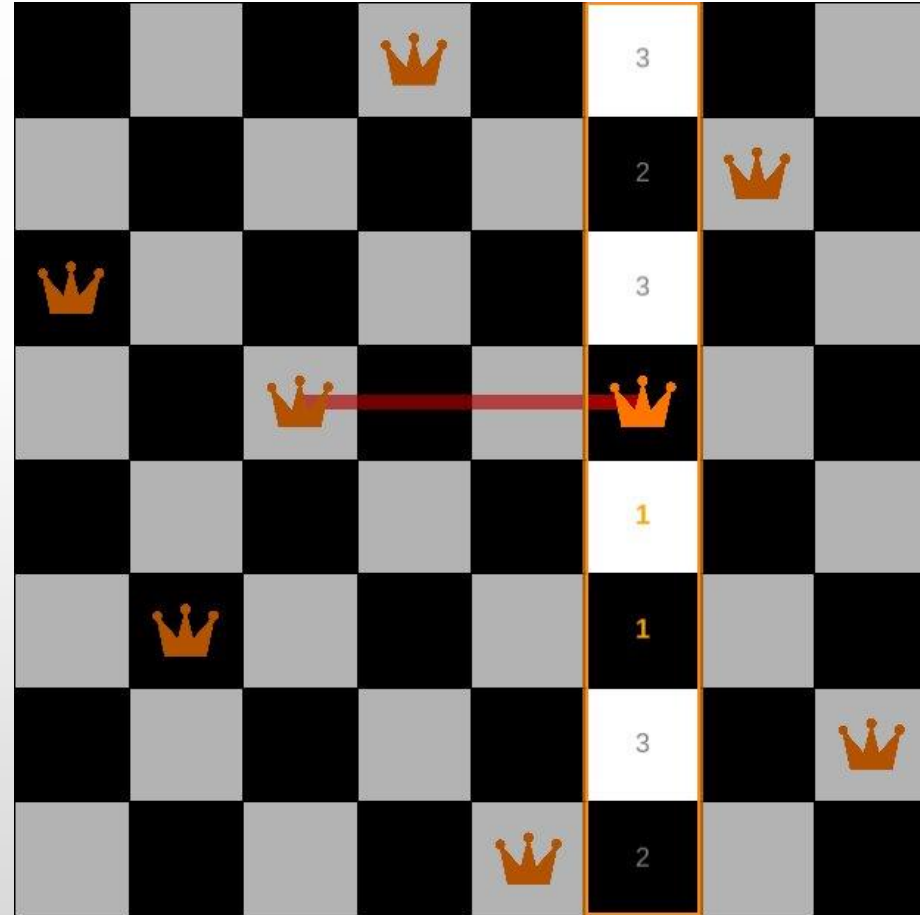
# 8 Queens



# 8 Queens

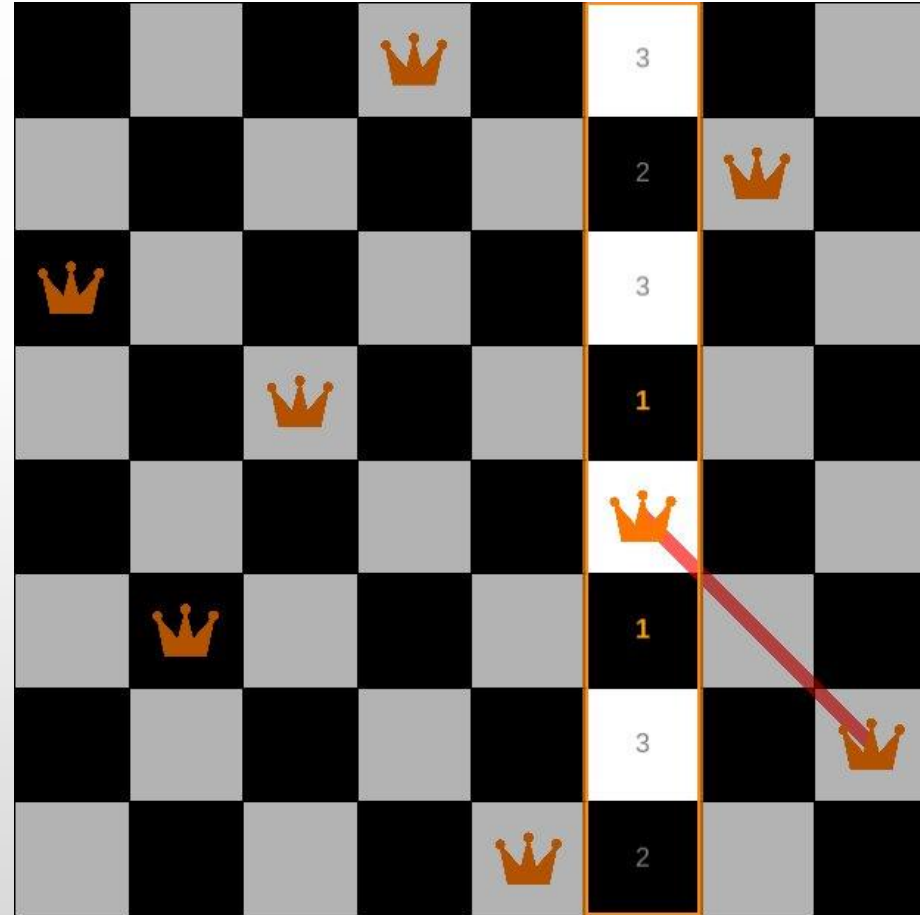


# 8 Queens

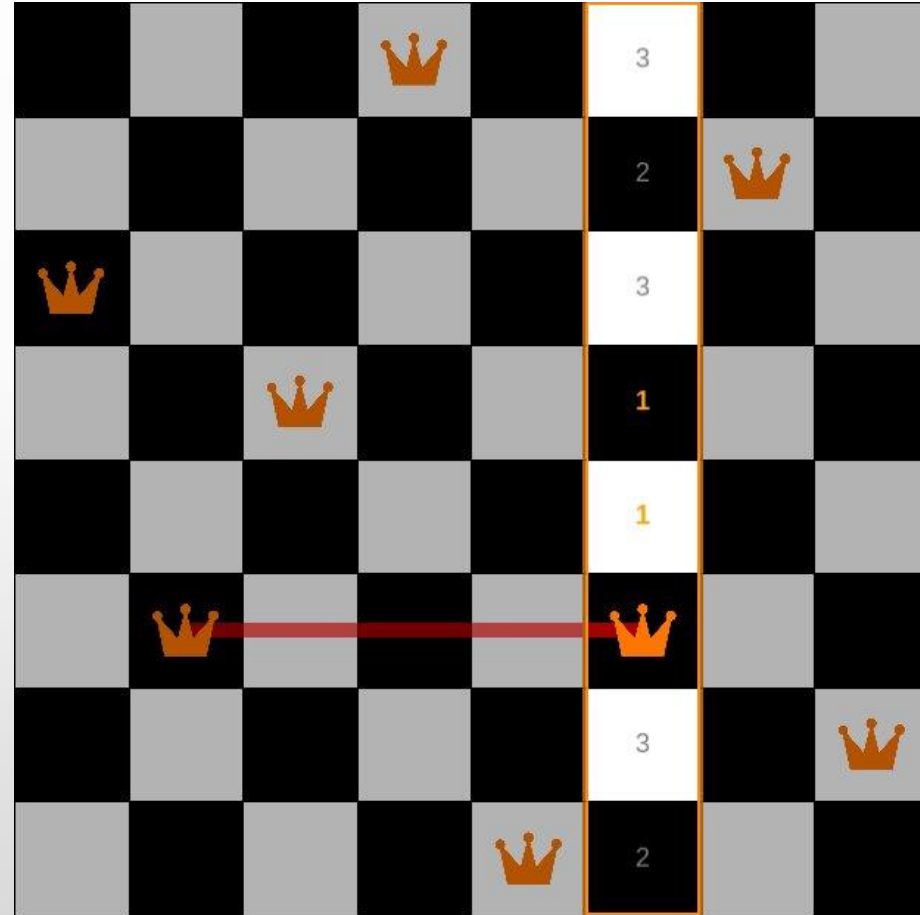




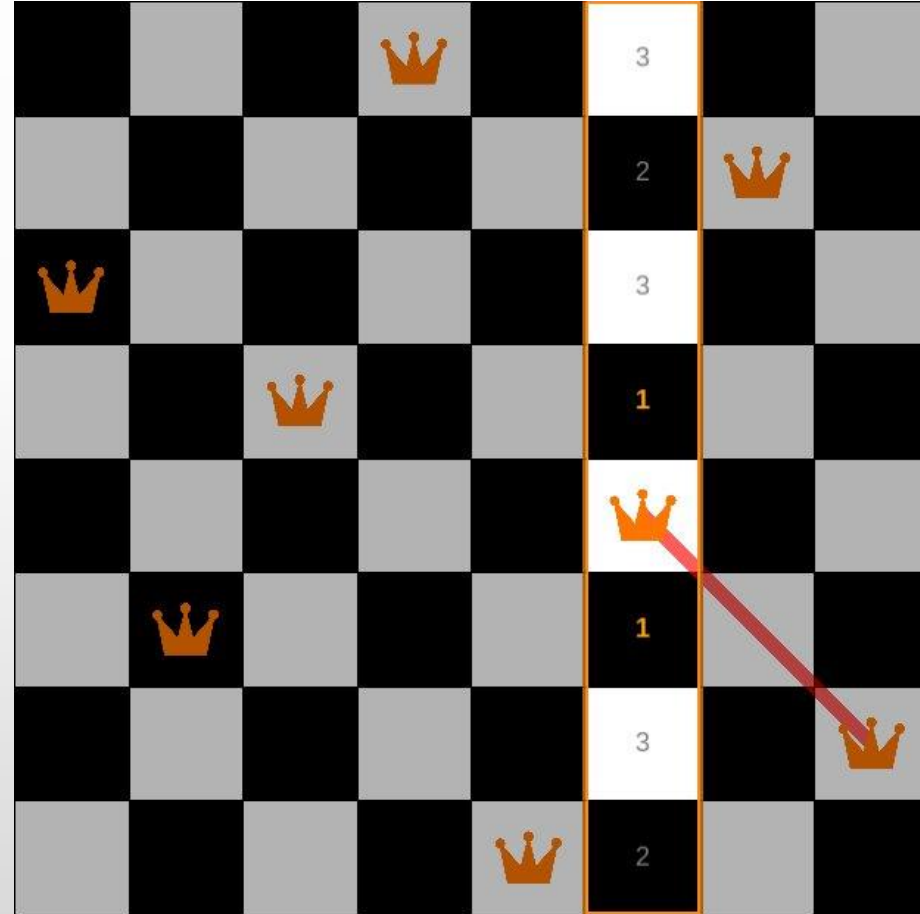
# 8 Queens



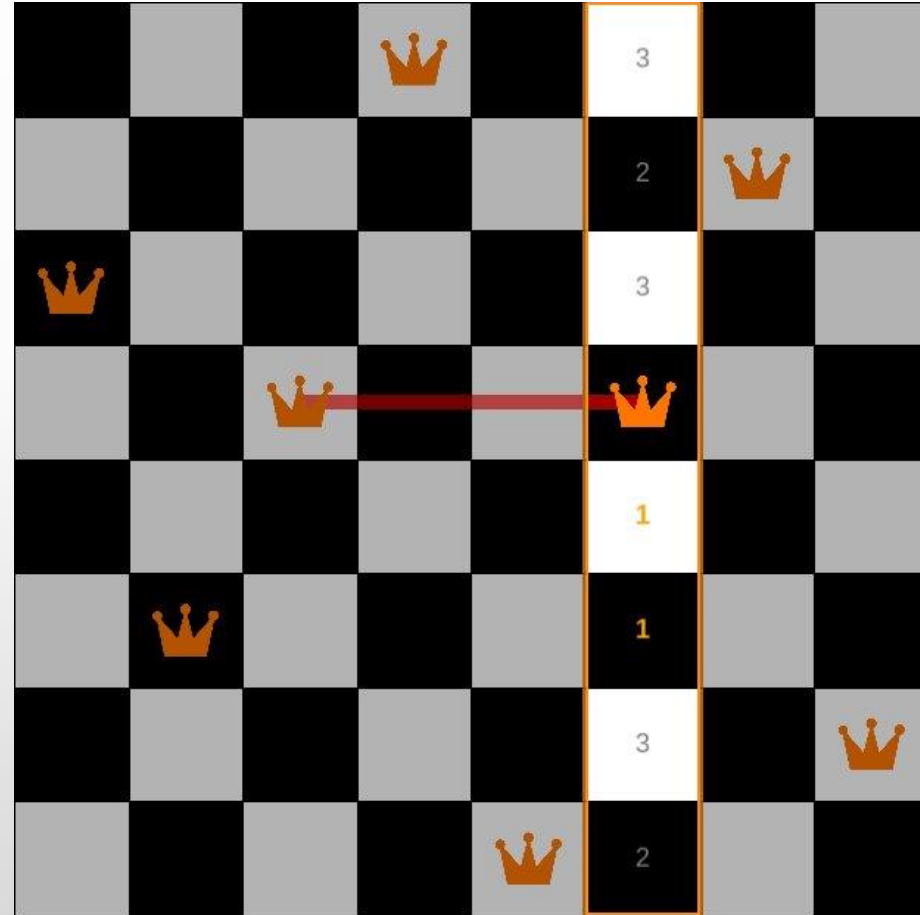
# 8 Queens



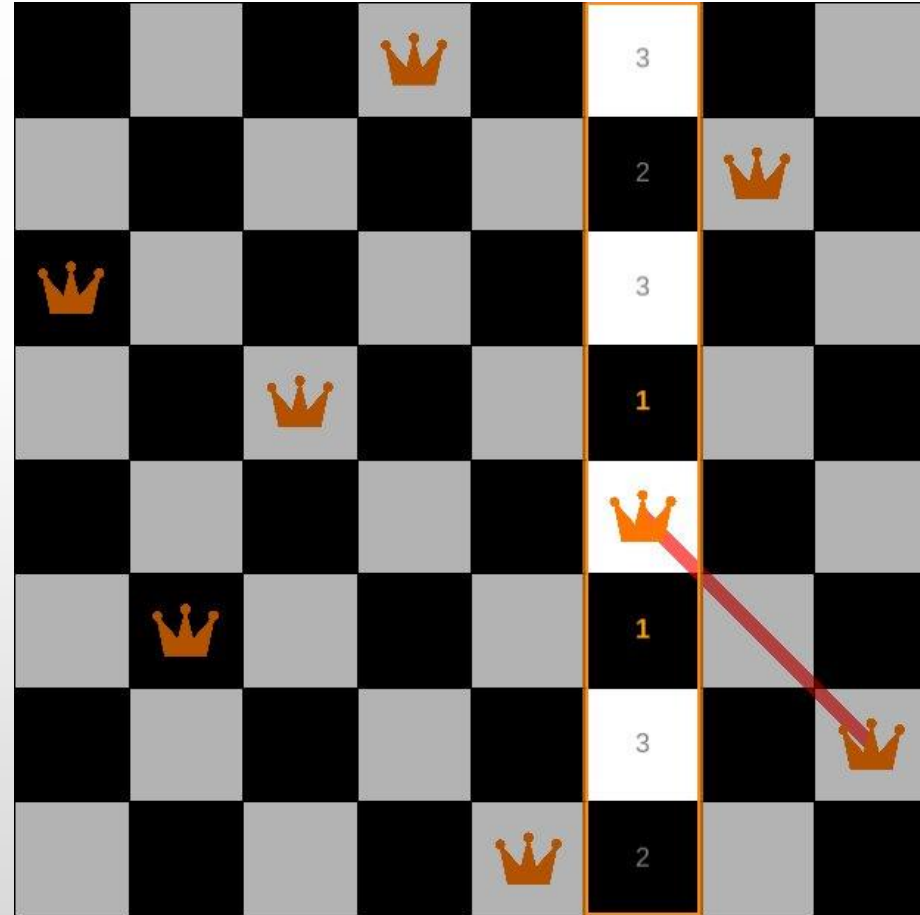
# 8 Queens



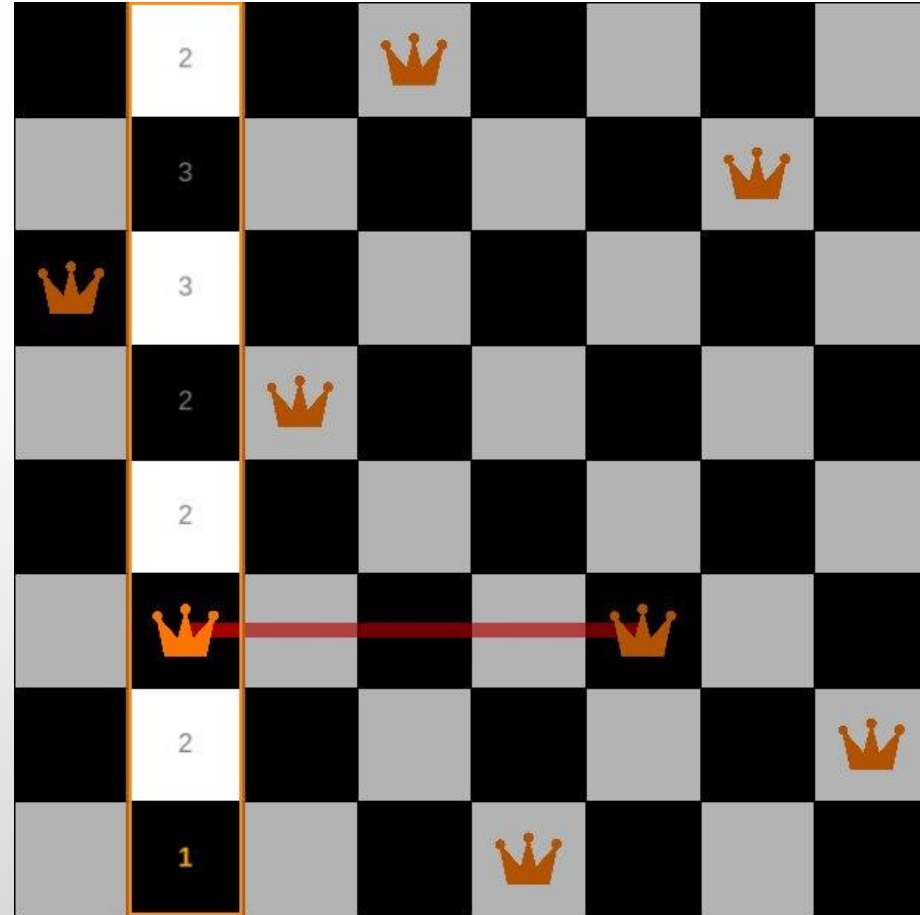
# 8 Queens



# 8 Queens

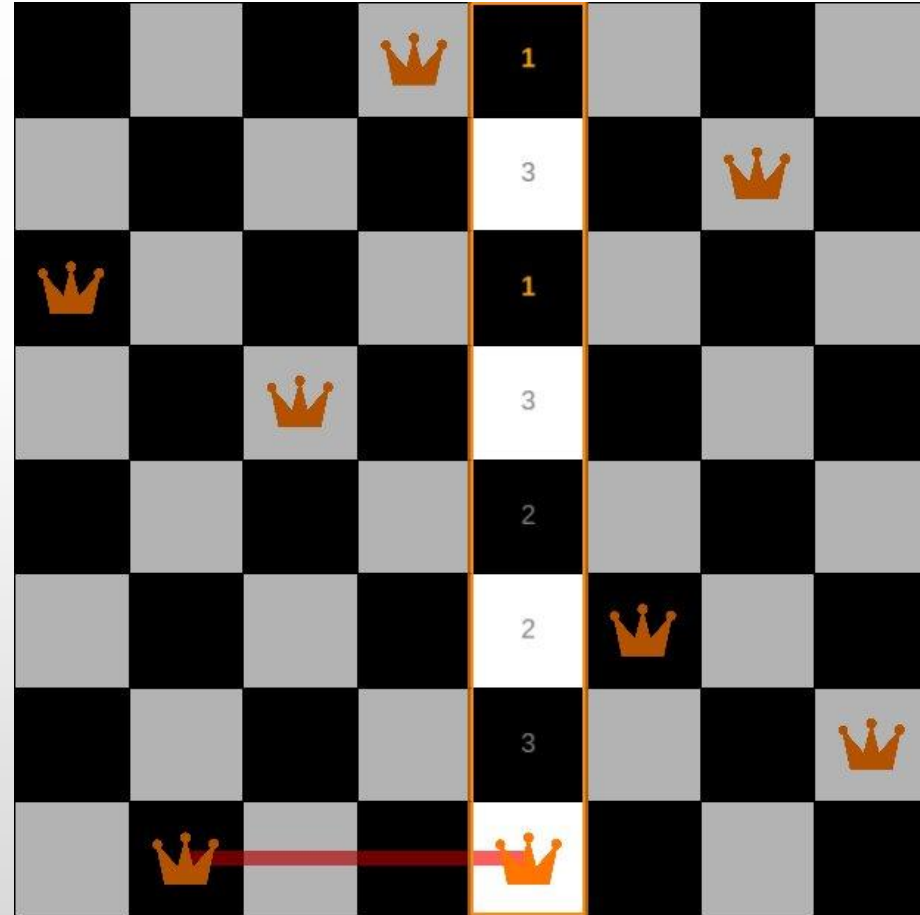


# 8 Queens

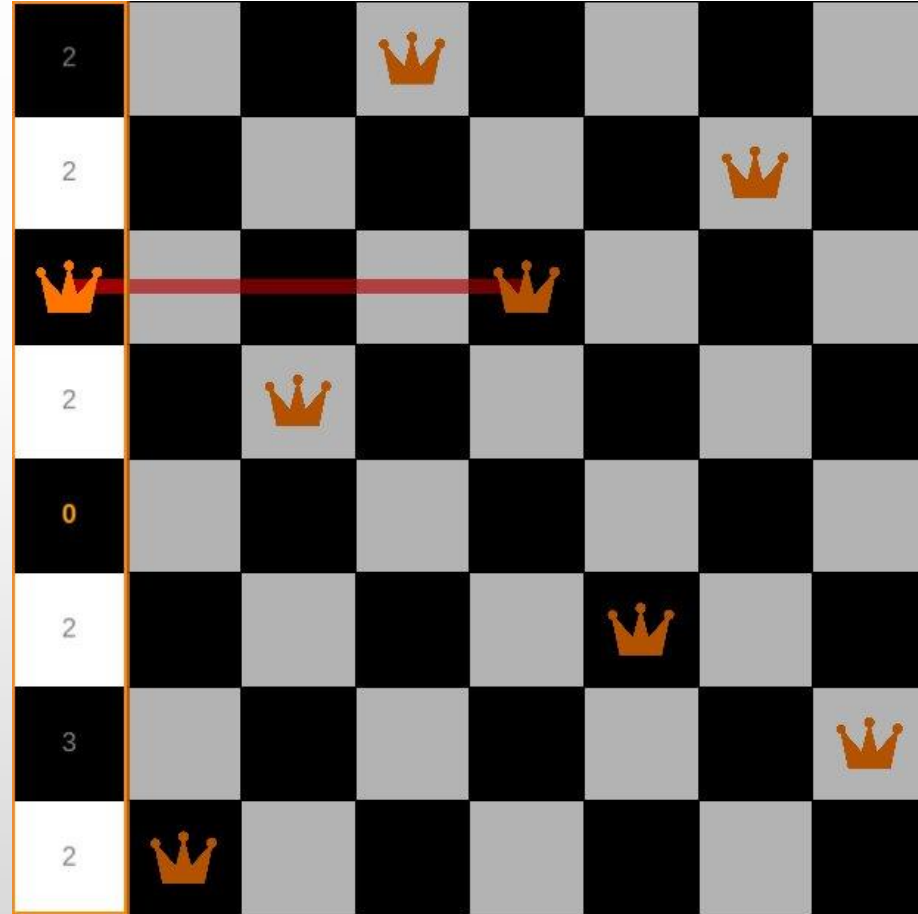




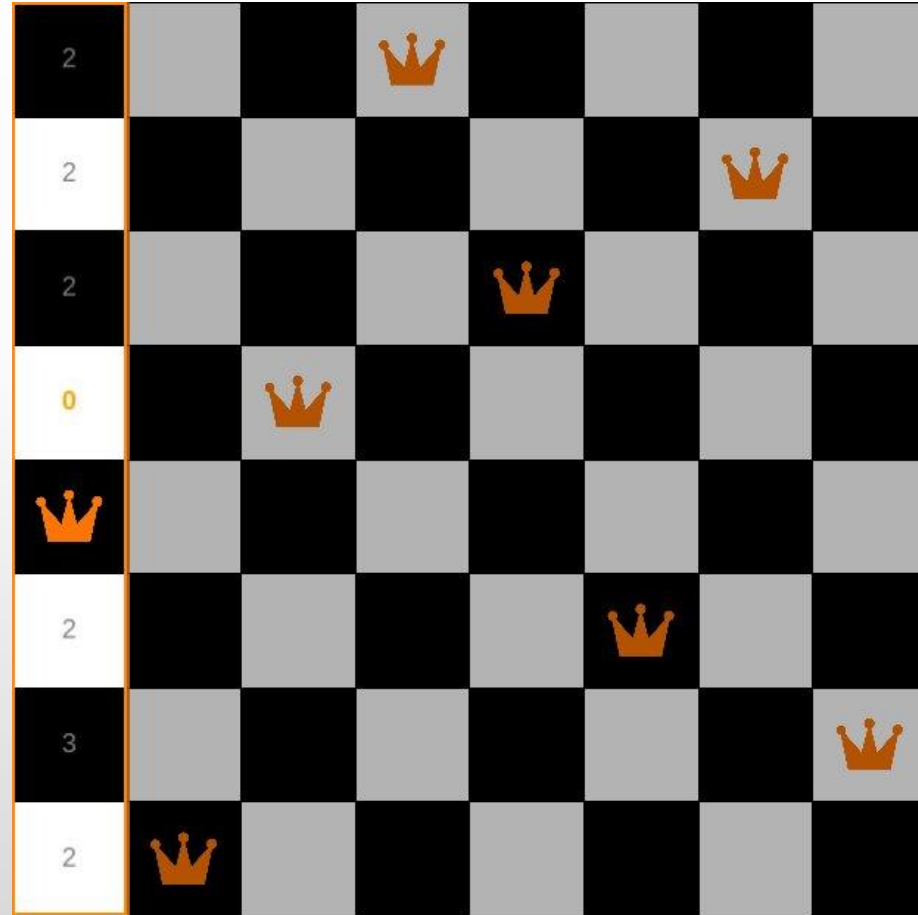
# 8 Queens



# 8 Queens



# 8 Queens





SON