

Bölüm 13: Geri Dönüşlü Algoritmalar Algoritmalar





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







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



- 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.



5	3	1	2	7	6	8	9	4
6	2	4	1	9	5	2		27.2
W. 100	9	8					6	0.00
8				6	1	- 2		3
4			8		3			9
7	3			2			11	6
8	6					2	8	
			4	9	9			5
(S)				8		- 30	7	9



5	3	1	2	7	6	9	4	8
6	4	2	1	9	5	7	3	4
Alexander :	9	8			-		6	0
8				6		- A1		3
4			8		3			9
7				2		13		6
89	6					2	8	
			4	9	9			5
(§	57			8		- 10	7	9



5	3	1	6	7	2	4	9	8
6	4	7	1	9	5	3	- 1	
	9	8		E 232 S			6	
000				6	1		120-3	3
4			8		3			1
7				2				6
80	6					2	8	
			4	9	9			5
68		3		8		1	7	9



5	3	1	6	7	2	8	9	4
6	7	4	1	9	5	6		2.0
States:	9	8					6	
8				6				3
4			8		3	39		9
7				2				6
83	6					2	8	
	7-		4	9	9			5
Ø				8		- 13	7	9



5	3	1	6	7	4	8	9	2
6	2	4	1	9	5	7	7	8
Ser inse	9	8				92	6	
8				6		1)(3
4			8		3	30		9
7				2				6
8	6					2	8	8
			4	9	9			5
82:		3		8		- 85	7	9



5	3	1	6	7	8	9	2	4
6	2	4	1	9	5	8	6	24
	9	8		8 / / / S			6	
8				6		W.		3
4			8		3	10		9
7				2				6
8	6					2	8	
			4	9	9	- Ingri		5
(Q		8		8		- 80	7	9



5	3	1	6	7	8	9	4	2
6	2	4	1	9	5	7	3	8
8	9	8				92	6	
8				6				3
4			8		3	- SS		9
7				2				6
8	6					2	8	
			4	9	9			5
13	82	3		8	- 4	- 40	7	9



(k)			4	1	9		7	5
	6					2	8	
7				2				6
4			8		3	33		9
000				6		U 5		3
Series	(P)	8		E 292 S	222.63	72	60	
60			9	9	5	8		22.
5	3	2		7	ė į:	- 20		60



5	3	2	6	7	8	1	9	4
6	4	7	1	9	5	3	2	8
1	9	8	2	2			6	SC
8				6				3
4			8		3	39		1
7		8		2				6
83	6					2	8	
			4	9	9	1000		5
Ø	83			8		10	7	9



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		l (3
4	V) M		8		3		***	1
7				2				6
80	6					2	8	
			4	1	9	YA.		5
£0	83			8			7	9



5	3	4	2	7	6	9	1	8
6	2	7	1	9	5	4	4	
States .	9	8	-w2				6	
8			Ÿ.	6				3
4	y et		8		3		***	1
7				2				6
	6					2	8	
			4	9	9	Y/.		5
20				8			7	9



5	3	4	6	7	2	8	9	3
6			1	9	5	- 8		27.
20120	9	8			2500 (%)	0.0	6	
8				6				3
4			8		3	- 30	E	1
7				2				6
	6					2	8	
			4	9	9			5
8		3		8		- 6	7	9



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	72 15400		9
7				2				6
88	6					2	8	
			4	9	9	. t g		5
£0	83			8		- 10	7	9



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	3	6		4	X	3
4	20		8		3	n Die		9
7		a		2				6
	6					2	8	
			4	9	9			5
()————	(5)			8		100	7	9



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	Op Op	8	2	3	4	5	6	7
8	1	5	9	6	1			3
4			8		3	39		1
7	9	3		2	02	- 3		6
	6		2			2	8	
			4	9	9			5
<u>(</u>	88			8			7	9



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	U-18-09		1
7		8		2				6
	6					2	8	
			4	1	9		121	5
	(8)			8		100	7	9



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	9
7	1	3	3	2				6
	6					2	8	
			4	9	9			5
Ø	88			8		- 18	7	9



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

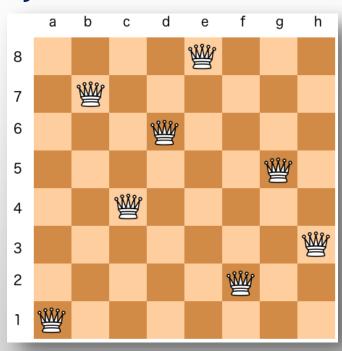


24

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

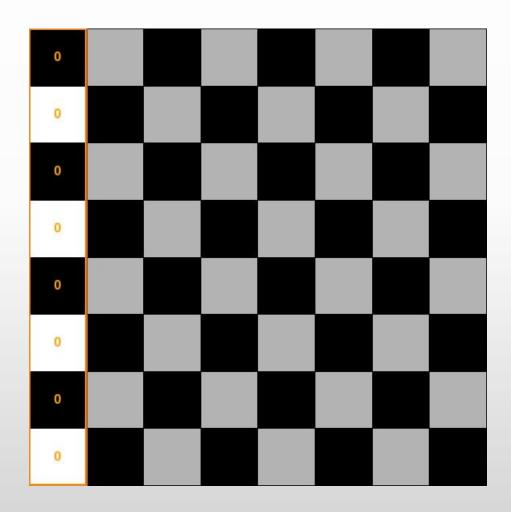


26

- 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.

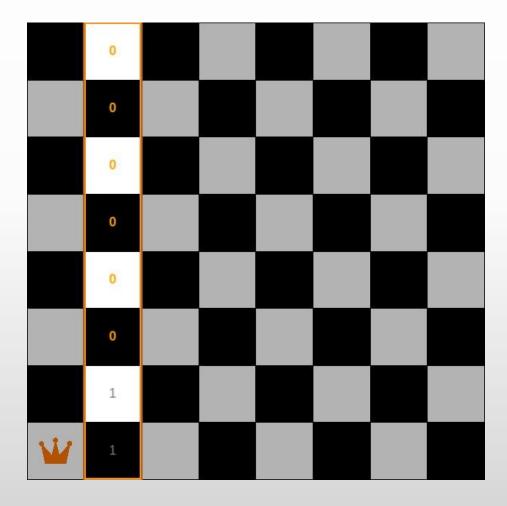








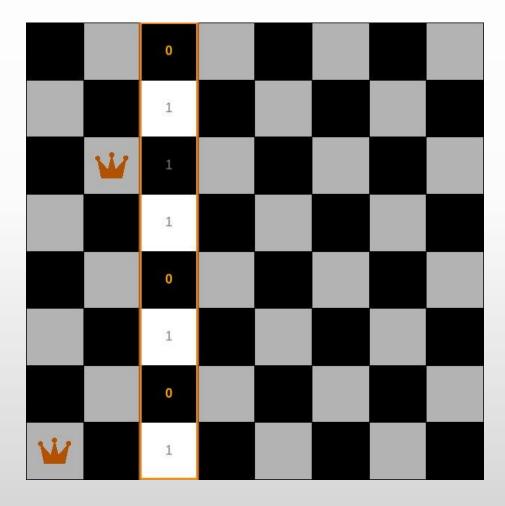




1/20/2023 Sercan KÜLCÜ, Tüm hakları saklıdır.



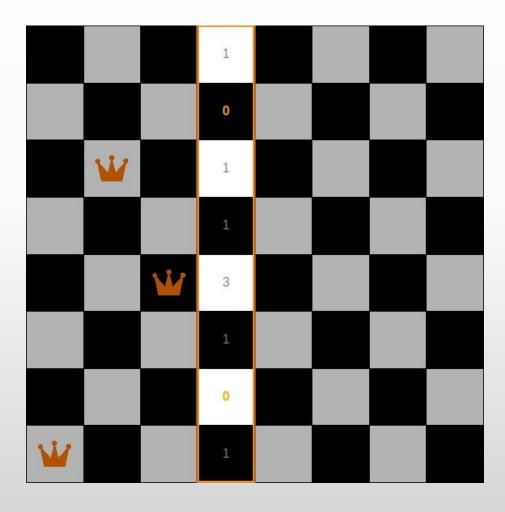




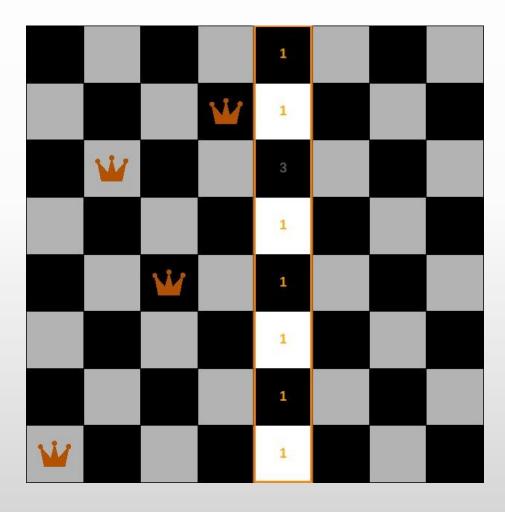
1/20/2023 Sercan KÜLCÜ, Tüm hakları saklıdır.



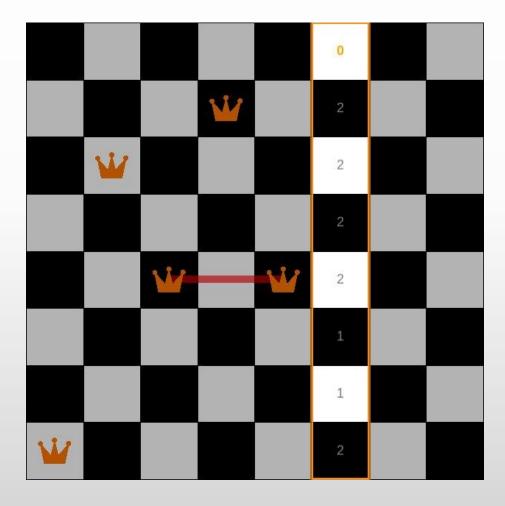






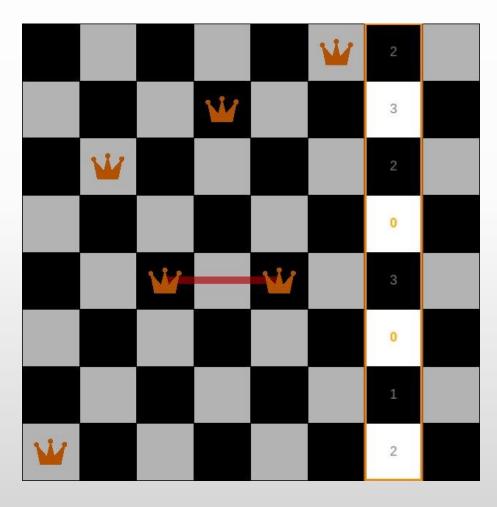






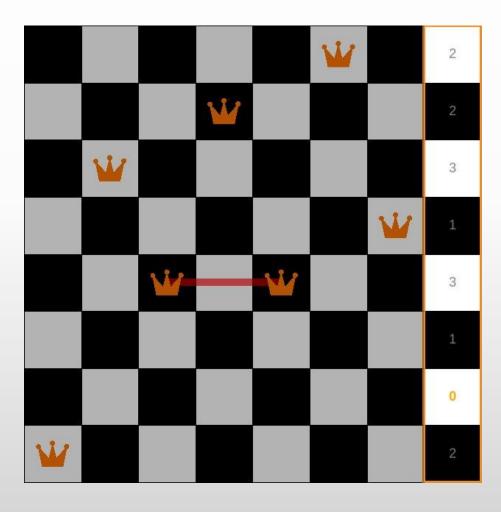
1/20/2023 Sercan KÜLCÜ, Tüm hakları saklıdır.



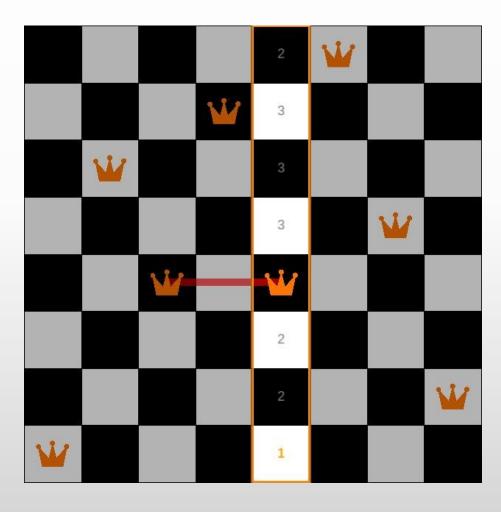


1/20/2023 Sercan KÜLCÜ, Tüm hakları saklıdır.

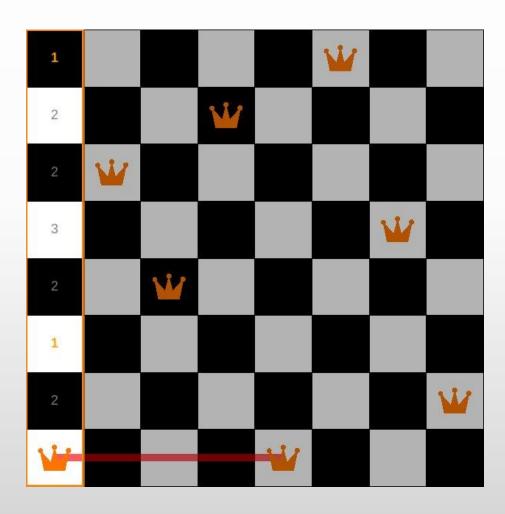




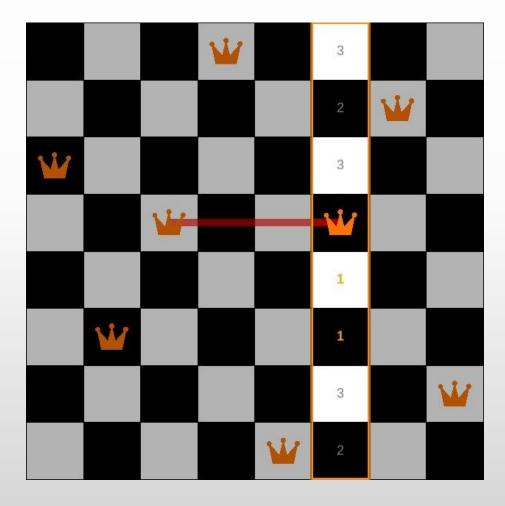








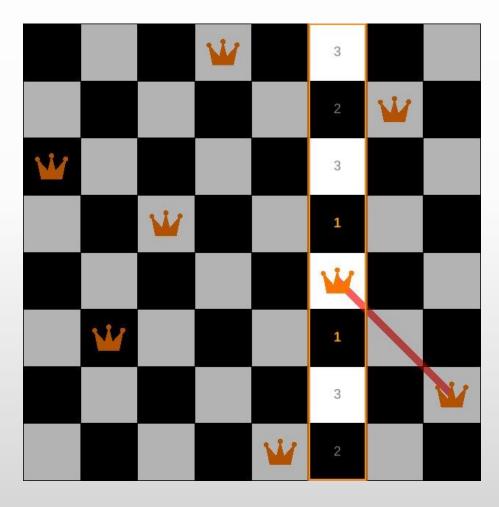




1/20/2023 Sercan KÜLCÜ, Tüm hakları saklıdır.

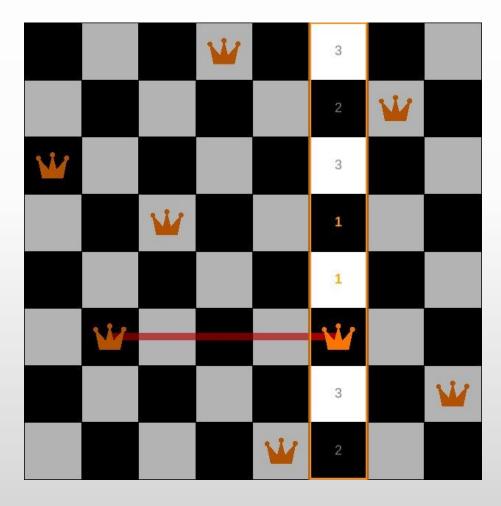




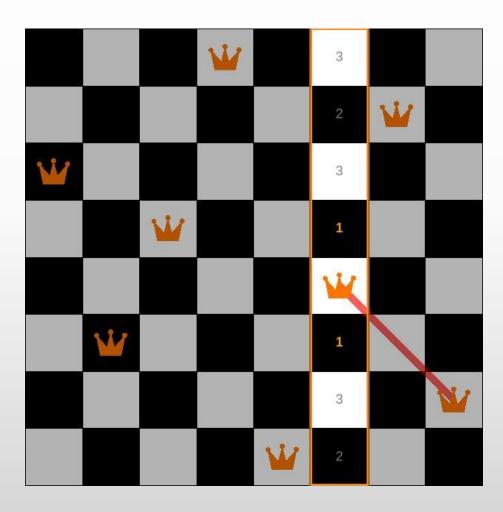




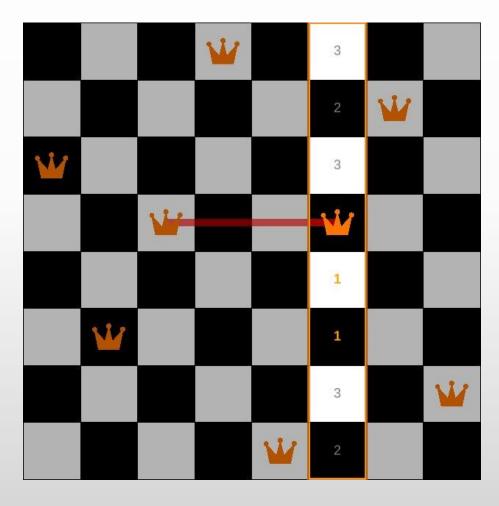






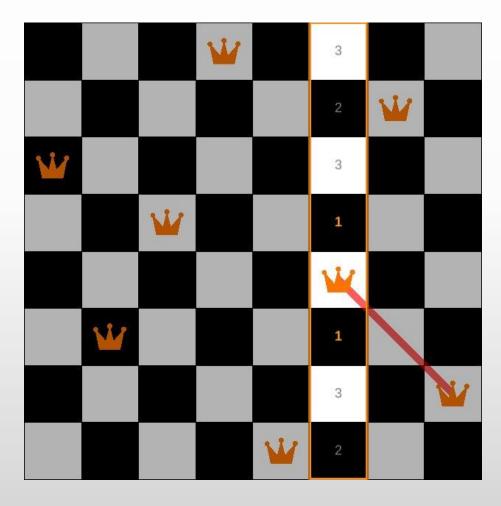






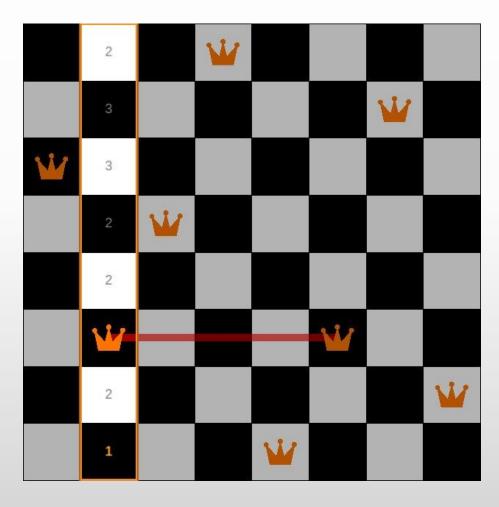






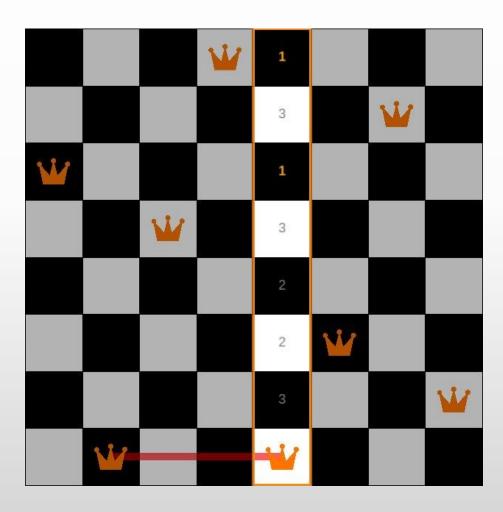




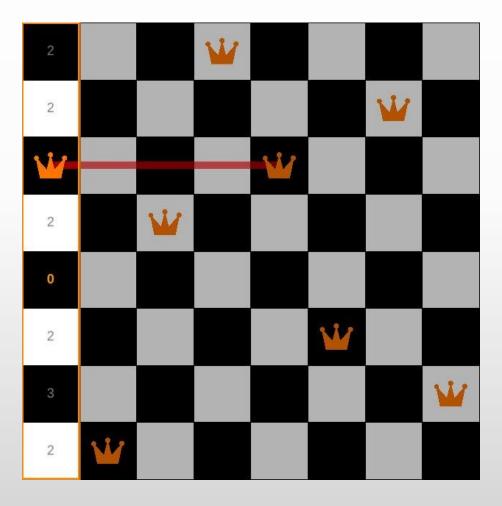








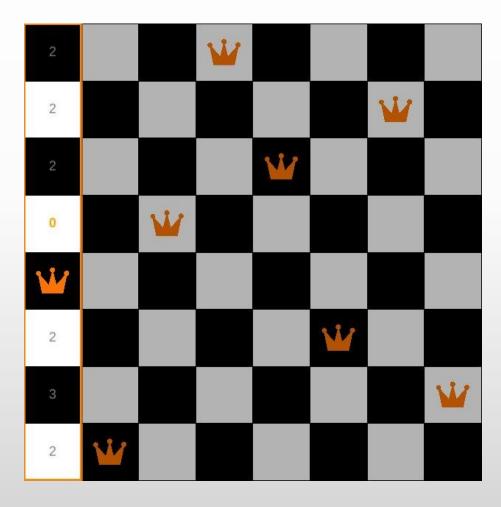




1/20/2023 Sercan KÜLCÜ, Tüm hakları saklıdır. 45









SON