Halil Burak Pala – 2019400282
Berat Damar - 2018400039

CMPE 300 ANALYSIS OF ALGORITHMS

PROJECT 3 – ANSWERS

PART 1

d) (You can adjust the length of the tables)

Success, n=6

Step	Columns	Available
1	[2]	[0, 4, 5]
2	[2, 5]	[0, 1, 3]
3	[2, 5, 1]	[4]
4	[2, 5, 1, 4]	[0]
5	[2, 5, 1, 4, 0]	[3]
6	[2, 5, 1, 4, 0, 3]	[]

Visualization of the table

	0	**1**	**2**	**3**	**4**	**5**
0			**Q**			
1						**Q**
2		**Q**				
3					**Q**	
4	**Q**					
5				**Q**		

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

Success, n=6

Step	Columns	Available
1	[1]	[3, 4, 5]
2	[1, 3]	[0, 5]
3	[1, 3, 5]	[0, 2]
4	[1, 3, 5, 0]	[2, 4]
5	[1, 3, 5, 0, 2]	[4]
6	[1, 3, 5, 0, 2, 4]	[]

Visualization of the table

	0	**1**	**2**	**3**	**4**	**5**
0		**Q**				
1				**Q**		
2						**Q**
3	**Q**					
4			**Q**			
5					**Q**	

Failure, n=6

Step	Columns	Available
1	[3]	[0, 1, 5]
2	[3, 5]	[0, 2]
3	[3, 5, 0]	[2, 4]
4	[3, 5, 0, 4]	[1]
5	[3, 5, 0, 4, 1]	[]

Failure, n=6

Step	Columns	Available
1	[1]	[3, 4, 5]
2	[1, 4]	[0, 2]
3	[1, 4, 0]	[3, 5]
4	[1, 4, 0, 3]	[]

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

Success, n=8

Step	Columns	Available
1	[5]	[0, 1, 2, 3, 7]
2	[5, 2]	[0, 4, 6]
3	[5, 2, 0]	[3, 6, 7]
4	[5, 2, 0, 7]	[3, 4]
5	[5, 2, 0, 7, 3]	[1]
6	[5, 2, 0, 7, 3, 1]	[3]
7	[5, 2, 0, 7, 3, 1, 6]	[6]
8	[5, 2, 0, 7, 3, 1, 6, 4]	[]

Visualization of the table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0						**Q**		
1			**Q**					
2	**Q**							
3								**Q**
4				**Q**				
5		**Q**						
6							**Q**	
7					**Q**			

Success, n=8

Step	Columns	Available
1	[5]	[0, 1, 2, 3, 7]
2	[5, 3]	[0, 1, 6]
3	[5, 3, 6]	[0, 4]
4	[5, 3, 6, 0]	[2, 7]
5	[5, 3, 6, 0, 2]	[4]
6	[5, 3, 6, 0, 2, 4]	[1, 7]
7	[5, 3, 6, 0, 2, 4, 1]	[7]
8	[5, 3, 6, 0, 2, 4, 1, 7]	[]

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

Visualization of the table

	0	1	2	3	4	5	6	7
0						**Q**		
1				**Q**				
2							**Q**	
3	**Q**							
4			**Q**					
5					**Q**			
6		**Q**						
7								**Q**

Failure, n=8

Step	Columns	Available
1	[2]	[0, 4, 5, 6, 7]
2	[2, 4]	[1, 6, 7]
3	[2, 4, 1]	[3, 7]
4	[2, 4, 1, 7]	[0, 5]
5	[2, 4, 1, 7, 5]	[3]
6	[2, 4, 1, 7, 5, 3]	[0, 6]
7	[2, 4, 1, 7, 5, 3, 0]	[]

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

Failure, n=8

Step	Columns	Available
1	[6]	[0, 1, 2, 3, 4]
2	[6, 0]	[2, 3, 5, 7]
3	[6, 0, 2]	[4, 5, 7]
4	[6, 0, 2, 5]	[1, 7]
5	[6, 0, 2, 5, 7]	[]

Success, n=10

Step	Columns	Available
1	[3]	[0, 1, 5, 6, 7, 8, 9]
2	[3, 6]	[0, 2, 4, 8, 9]
3	[3, 6, 0]	[2, 5, 7, 9]
4	[3, 6, 0, 5]	[1, 8]
5	[3, 6, 0, 5, 1]	[4, 9]
6	[3, 6, 0, 5, 1, 9]	[7]
7	[3, 6, 0, 5, 1, 9, 7]	[2]
8	[3, 6, 0, 5, 1, 9, 7, 2]	[4, 8]
9	[3, 6, 0, 5, 1, 9, 7, 2, 4]	[8]
10	[3, 6, 0, 5, 1, 9, 7, 2, 4, 8]	[]

Visualization of the table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0				**Q**						
1							**Q**			
2	**Q**									
3						**Q**				
4		**Q**								
5										**Q**
6								**Q**		
7			**Q**							
8					**Q**					
9									**Q**	

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

Success, n=10

Step	Columns	Available
1	[7]	[0, 1, 2, 3, 4, 5, 9]
2	[7, 0]	[2, 3, 4, 6, 8]
3	[7, 0, 3]	[1, 5, 6, 8, 9]
4	[7, 0, 3, 8]	[2, 4, 6]
5	[7, 0, 3, 8, 6]	[1, 9]
6	[7, 0, 3, 8, 6, 9]	[2]
7	[7, 0, 3, 8, 6, 9, 2]	[5]
8	[7, 0, 3, 8, 6, 9, 2, 5]	[1]
9	[7, 0, 3, 8, 6, 9, 2, 5, 1]	[4]
10	[7, 0, 3, 8, 6, 9, 2, 5, 1, 4]	[]

Visualization of the table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0								**Q**		
1	**Q**									
2				**Q**						
3									**Q**	
4							**Q**			
5										**Q**
6			**Q**							
7						**Q**				
8		**Q**								
9					**Q**					

Failure, n=10

Step	Columns	Available
1	[8]	[0, 1, 2, 3, 4, 5, 6]
2	[8, 0]	[2, 3, 4, 5, 7, 9]
3	[8, 0, 9]	[1, 3, 4, 6, 7]
4	[8, 0, 9, 3]	[1, 5, 6]
5	[8, 0, 9, 3, 1]	[7]
6	[8, 0, 9, 3, 1, 7]	[4]
7	[8, 0, 9, 3, 1, 7, 4]	[2]
8	[8, 0, 9, 3, 1, 7, 4, 2]	[]

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

Failure, n=10

Step	Columns	Available
1	[9]	[0, 1, 2, 3, 4, 5, 6, 7]
2	[9, 0]	[2, 3, 4, 5, 6, 8]
3	[9, 0, 2]	[4, 5, 7, 8]
4	[9, 0, 2, 4]	[1, 6, 7, 8]
5	[9, 0, 2, 4, 7]	[1, 3]
6	[9, 0, 2, 4, 7, 3]	[8]
7	[9, 0, 2, 4, 7, 3, 8]	[]

d)

n	Number of Success	Number of Trials	Probability
6	732	10000	0.0732
8	1317	10000	0.1317
10	624	10000	0.0624

PART 2

c)

n = 6

k	Number of Success	Number of Trials	Probability
0	10000	10000	1.0
1	6629	10000	0.6629
2	2217	10000	0.2217
3	1152	10000	0.1152
4	835	10000	0.0835
5	856	10000	0.0856

n = 8

k	Number of Success	Number of Trials	Probability
0	10000	10000	1.0
1	10000	10000	1.0
2	8749	10000	0.8749
3	4998	10000	0.4998
4	2641	10000	0.2641
5	1708	10000	0.1708
6	1921	10000	0.1921
7	3131	10000	0.3131

Halil Burak Pala – 2019400282
Berat Damar - 2018400039

n = 10

k	Number of Success	Number of Trials	Probability
0	10000	10000	1.0
1	10000	10000	1.0
2	10000	10000	1.0
3	7997	10000	0.7997
4	4177	10000	0.4177
5	1936	10000	0.1936
6	1165	10000	0.1165
7	956	10000	0.0956
8	1222	10000	0.1222
9	1946	10000	0.1946

d) Comments

For n=6, when k=0, success rate is 100% since in that case it is a pure deterministic algorithm. Then for k values greater than 0, success rate decreases until k=5, then for k=5 it increases.

For n=8, again, when k=0, success rate is 100% since it is a pure deterministic algorithm. But success rate is 100% for k=1 also. That is because for every column in the first row, there is a solution for this problem and our deterministic algorithm can find every solution. For k values that are greater than 1, success rate decreases until k=6, then it increases for k=6 and k=7.

For n=10, success rate is 100% for k=0 since it is a pure deterministic algorithm. For k=1 and k=2, success rate is again 100%. Because for every column in the first row, we can find a solution to this problem, and this is also the case for every legitimate two column combinations for first two rows. For k values greater than 2, success rate decreases until k=8, then it increases for k=8 and k=9.

The reason for decreasing until a specific k that is close to n and then increasing is that while adding new queens to columns, we eliminate some other correct variations of placements, and we reduce number of solutions. But after we place some specific number of queens that is close to n, we are closer to reaching a specific solution with higher probability.

When we compare the success rates of these two algorithms, we can see that using backtracking after placing some number of queens with a Las Vegas algorithm gives higher success rates than a pure Las Vegas algorithm. This is because backtracking algorithm is a deterministic algorithm, so if there is a solution after placing some number of queens, deterministic algorithm finds this solution surely. So, this increases the probability of success.