Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

CMPE 300 ANALYSIS OF ALGORITHMS

PROJECT 3 - ANSWERS

PART 1

d)

Success, n=6

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

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

Success, n=6

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

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	[2]	[0, 4, 5]
2	[2, 0]	[3, 5]
3	[2, 0, 3]	[1]
4	[2, 0, 3, 1]	[4]
5	[2, 0, 3, 1, 4]	[]

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, 2]	[4]
5	[3, 5, 0, 2, 4]	[]

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

Success, n=8

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

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**					

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

Success, n=8

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

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, 7]	[0, 1, 3]
4	[2, 4, 7, 1]	[3]
5	[2, 4, 7, 1, 3]	[5, 6]
6	[2, 4, 7, 1, 3, 5]	[0]
7	[2, 4, 7, 1, 3, 5, 0]	[]

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

Failure, n=8

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

Success, n=10

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

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

Success, n=10

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

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

Failure, n=10

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

d)

n	Number of Success	Number of Trials	Probability
6	693	10000	0.0693
8	1297	10000	0.1297
10	626	10000	0.0626

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

PART 2

c)

n = 6

k	Number of Success	Number of Trials	Probability
0	10000	10000	1.0
1	6623	10000	0.6623
2	2185	10000	0.2185
3	1112	10000	0.1112
4	812	10000	0.0812
5	832	10000	0.0832

n = 8

k	Number of Success	Number of Trials	Probability
0	10000	10000	1.0
1	10000	10000	1.0
2	8736	10000	0.8736
3	4983	10000	0.4983
4	2642	10000	0.2642
5	1749	10000	0.1749
6	1882	10000	0.1882
7	3032	10000	0.3032

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

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	8022	10000	0.8022
4	4137	10000	0.4137
5	1964	10000	0.1964
6	1118	10000	0.1118
7	957	10000	0.0957
8	1199	10000	0.1199
9	2041	10000	0.2041

d) Comments

It can be seen from the experimental results that as **k** increases, probability of success decreases for a while. For n=6, k=0 gives 100% probability of success. It is quite natural since we do not introduce any randomization to the algorithm yet, we solve the whole problem with deterministic approach. When k=1, probability of success is not 100%, which shows that in order to achieve success from the very beginning we should be careful about our selection of the column. Other important observation is that probability of success increases after a point. When n=6, probability of success with k=5 is greater than probability of success with k=4. It wouldn't be the case if we didn't rerun the probabilistic part of the algorithm in order to place first k queens without failure. Indeed if we didn't rerun, we would get a probability of success similar to what we get in the first part since we get close to the full-probabilistic approach. Rerunning the probabilistic part in case of a failure compensates some of the erroneous cases and therefore leads to a better probability of success.

For n=8, k=0 and k=1 gives 100% probability of success. It shows that we can select the first column for the queen arbitrarily. For any selection, there exists at least one solution. Therefore, we still have 100% probability of success given the fact that we solve the remaining part of the question deterministically. However, it's not the case when k ≥ 2. If we select first two queens randomly, then we introduce the probability of failure. Similar to n=6 case, as k increases probability of success decreases for a while. For k=6 and k=7, probability of success starts increasing.

For n=10, k=0, k=1 and k=2 gives 100% probability of success. It's interesting to see that when n=10, we can locate first two queens randomly and still be able to solve the problem. When k ≥ 3, introduction of randomization begins to show its effects. Probability of success starts increasing when k=8, again due to the reason mentioned above.

Furkan Özdemir – 2018400201
Karahan Sarıtaş - 2018400174

Overall, introduction of randomization reduces the probability of success up to a point (as expected). Rerunning the probabilistic part compensates some of the failures and increases the probability of success after a point. It’s because after selecting a specific number of k queens successfully (in the probabilistic part), probability of encountering with a dead-end in the deterministic part decreases, therefore probability of success increases. Keep in mind that it requires lots of rerunning in the background. Randomization also reduces the time it requires to complete the algorithm. Please try running part2 and observe the difference between time it takes for n = 8, k = 0 and n = 8, k = 7. Therefore for larger n values, it may become favorable to use a probabilistic algorithm several times until it comes to a solution rather than implementing a deterministic algorithm. Additionally, rerunning the probabilistic part increases the time it requires for algorithm to complete for larger k values, since number of dead-ends that will be encountered during the probabilistic part increases. Therefore there is also a tradeoff between a ‘rerunning probabilistic part’ approach and allowing failures in the probabilistic part. Second case would give probability of success results similar to the results we get in the first part when k gets closer to n, however that would require less time than the first approach.