CMPE 300 ANALYSIS OF ALGORITHMS
PROJECT 3 - ANSWERS

PART 1

1.1) Fill the steps of one successful and one unsuccessful execution for each p value

1.1.1) Success - p=0.7

	0	1	2	3	4	5	6	7
0	-1	36	29	4	19	-1	27	-1
1	30	3	6	37	28	-1	-1	17
2	35	-1	-1	20	5	18	39	26
3	-1	31	2	7	38	25	16	-1
4	-1	34	21	-1	15	8	-1	40
5	32	1	14	-1	24	43	12	9
6	-1	22	33	0	13	10	41	44
7	-1	-1	-1	23	42	45	-1	11

1.1.2) Unsuccessful - p=0.7

	0	1	2	3	4	5	6	7
0	12	33	10	-1	-1	31	-1	-1
1	-1	-1	13	32	-1	-1	-1	30
2	34	11	-1	9	14	29	0	-1
3	-1	8	3	-1	1	18	15	-1
4	4	35	6	19	28	39	-1	17
5	7	-1	21	2	-1	16	25	40
6	-1	5	36	23	20	27	38	-1
7	-1	22	-1	-1	37	24	41	26

1.1.3) Success - p=0.8

	0	1	2	3	4	5	6	7
0	46	43	-1	31	40	-1	4	13
1	1	30	45	42	3	14	39	-1
2	44	47	2	15	32	41	12	5
3	29	0	33	-1	-1	52	-1	38
4	48	21	28	53	16	-1	6	11
5	27	18	25	34	51	8	37	-1
6	22	49	20	17	24	35	10	7
7	19	26	23	50	9	-1	-1	36

1.1.4) Unsuccessful - p=0.8

	0	1	2	3	4	5	6	7
0	-1	-1	-1	-1	-1	27	-1	-1
1	-1	-1	-1	24	-1	0	-1	28
2	-1	23	-1	-1	26	5	-1	-1
3	-1	2	25	4	-1	-1	11	6
4	22	-1	20	-1	-1	-1	14	-1
5	19	-1	3	-1	15	12	7	10
6	-1	21	-1	17	-1	9	-1	13
7	-1	18	-1	-1	-1	16	-1	8

1.1.5) Success - p=0.85

	0	1	2	3	4	5	6	7
0	-1	55	18	11	42	13	44	1
1	-1	10	53	14	21	0	41	-1
2	54	19	28	17	12	43	2	45
3	27	52	9	20	15	22	7	40
4	34	29	16	-1	8	-1	46	3
5	51	26	35	30	23	4	39	6
6	36	33	24	49	38	31	-1	47
7	25	50	37	32	-1	48	5	-1

1.1.6) Unsuccessful - p=0.85

	0	1	2	3	4	5	6	7
0	-1	-1	-1	-1	-1	24	-1	-1
1	-1	-1	8	25	-1	21	-1	23
2	-1	-1	-1	-1	9	26	-1	-1
3	-1	-1	0	7	20	-1	22	27
4	1	-1	-1	10	-1	16	13	-1
5	-1	-1	6	3	12	19	28	15
6	-1	2	11	-1	5	14	17	30
7	-1	-1	4	-1	18	29	-1	-1

1.2) Fill the table and comment on it

1.2.1)

p	Number of Success	Number of Trials	Probability	Total Time of Execution
0.7	16517	100000	% 16.517	54.015 s
0.8	2439	100000	% 2.439	54.429 s
0.85	577	100000	% 0.577	54.991 s

1.2.1) Comments

Also answer these questions while commenting

1- How do changes on p affect total success probability and total execution time?

2- Define trade-offs of the algorithm.

Increasing p reduces the success probability because basically the task becomes harder as we increase the number of spots to travel. So, we can say there is a trade-off between p-value and success rate. The reason for this is as we increase the p-value we have to make more random choices so, the probability to face an unsuccessful situation increases. Also, as we have more steps, number of empty slots decrease.
As the p-value increases the execution time gets shorter. Although the difference is not huge, there is a difference. The reason of this can be seen looking at the success rates. Normally, higher p-value takes more time in a successful case because we have to make more moves and we also know that unsuccessful cases should take less time than successful case. Comparing these two ideas, we can say that the superiority of unsuccessful cases have covered for the higher operation numbers in this case.

PART 2

2.1) Fill the tables and comment on them

2.1.1) p = 0.7

k	Number of Success	Number of Trials	Probability	Total Time
0	100000	100000	% 100	33.361 s
2	100000	100000	% 100	33.515 s
3	99929	100000	% 99.929	32.299 s

2.1.2) p = 0.8

k	Number of Success	Number of Trials	Probability	Total Time
0	100000	100000	% 100	38.114 s
2	100000	100000	% 100	40.069 s
3	99948	100000	% 99.948	48.375 s

2.1.1) p = 0.85

k	Number of Success	Number of Trials	Probability	Total Time
0	100000	100000	% 100	95.455 s
2	100000	100000	% 100	193.661 s
3	99939	100000	% 99.939	494.433 s

2.1.2) Comments

Also answer these questions while commenting

1- How does total time change with k?

2- How do total time change with p for a specific k value? How does this change different from the first part?

3- Run this algorithm for each p value for k a value larger than 10 multiple times. What are your thoughts?

As k increases the time for the algorithm increases because as we make more random choices we have to make more deterministic moves in order to compensate for the random choices.
As p increases the execution time increases again, the reason for this is higher p means, the algorithm needs to make longer travels and has to be more clever. Therefore the algorithm makes more backtraces and the time increases.
We ran the algorithm 100 times for k=12. Comparing to k=0, 1 and 2 cases k=12 was less successful as expected because of the high k value as we have just explained.

PART 3

In this part, you will compare Part1 and Part2 algorithms according to their ability to solve the actual Knight's Problem where p=1.

- Run Part1 algorithm with p=1.
- Run Part2 algorithm with p=1 and k=0.
- Run Part2 algorithm with p=1 and a k value you think will work well.

Clearly state your findings and comment on them. When would you choose Part1 algorithm and when would you choose the other?

- When we ran the algorithm 100 000 times with p=1, we received a output with 0 successful tours. This is an expected result because travelling the whole table is a very rare case among all the possibilities and our code could not achieve to do it once in 100 thousand totally random tries.

- When we ran Part2 with p=1 and k=0, our program did not give an output in a feasible time interval.
- We tried 12 and 20 as k values & they did not give output in a reasonable time too. So, we tried 32 as the k value and it needed a very long time to give output. The result was 83 successful cases in 100 000 tries. When we made k = 40, we got output in very reasonable time with only 5 successful cases out of 100 thousand. After we tried 50 and we got 0.

Considering these two cases together, we can say deterministic approach takes so much time against the probabilistic approach because it tries all the possible cases before deciding it has failed. As we increase the k to really high values. The algorithm starts to fail in the probabilistic part, so it does not have very long deterministic session and we can get output in more reasonable times. And obviously, success rates decrease to very low values because we make so many random choices.