

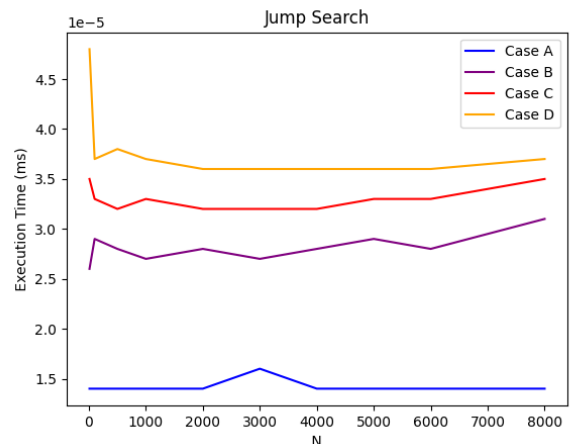
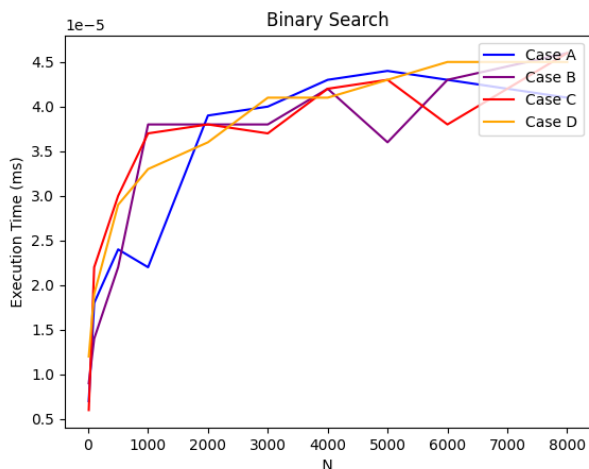
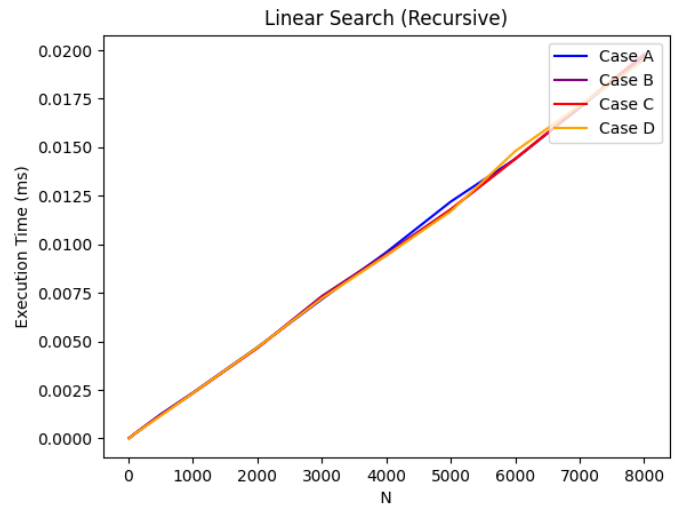
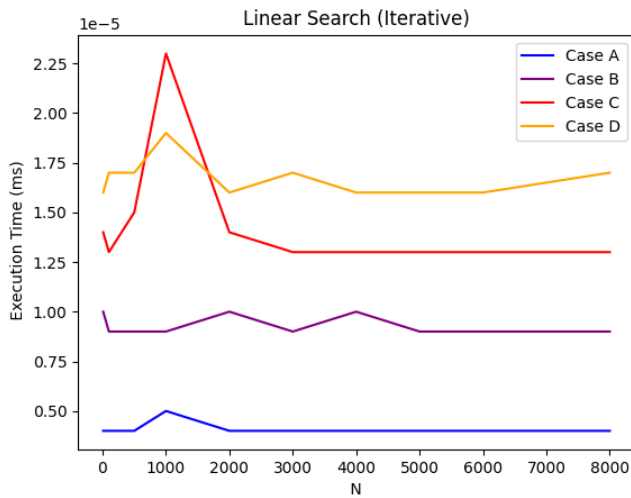
SPRING 2020-21 CS 201 HOMEWORK ASSIGNMENT 02

Sarper Turan 21903051 CS 201-01

3) –the unit of time values are milliseconds throughout the homework-

N	Linear Search (Iterative)				Linear Search (Recursive)				Binary Search				Jump Search			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
10	4,00E-06	1,00E-05	1,40E-05	1,60E-05	2,30E-05	1,20E-05	7,00E-06	3,00E-06	7,00E-06	9,00E-06	6,00E-06	1,20E-05	1,40E-05	2,60E-05	3,50E-05	4,80E-05
100	4,00E-06	9,00E-06	1,30E-05	1,70E-05	2,61E-04	2,50E-04	2,49E-04	2,55E-04	1,80E-05	1,40E-05	2,20E-05	1,90E-05	1,40E-05	2,90E-05	3,30E-05	3,70E-05
500	4,00E-06	9,00E-06	1,50E-05	1,70E-05	1,24E-03	1,23E-03	1,19E-03	1,17E-03	2,40E-05	2,20E-05	3,00E-05	2,90E-05	1,40E-05	2,80E-05	3,20E-05	3,80E-05
1000	5,00E-06	9,00E-06	2,30E-05	1,90E-05	2,36E-03	2,34E-03	2,33E-03	2,34E-03	2,20E-05	3,80E-05	3,70E-05	3,30E-05	1,40E-05	2,70E-05	3,30E-05	3,70E-05
2000	4,00E-06	1,00E-05	1,40E-05	1,60E-05	4,70E-03	4,66E-03	4,66E-03	4,69E-03	3,90E-05	3,80E-05	3,80E-05	3,60E-05	1,40E-05	2,80E-05	3,20E-05	3,60E-05
3000	4,00E-06	9,00E-06	1,30E-05	1,70E-05	7,17E-03	7,33E-03	7,26E-03	7,21E-03	4,00E-05	3,80E-05	3,70E-05	4,10E-05	1,60E-05	2,70E-05	3,20E-05	3,60E-05
4000	4,00E-06	1,00E-05	1,30E-05	1,60E-05	9,59E-03	9,52E-03	9,43E-03	9,43E-03	4,30E-05	4,20E-05	4,20E-05	4,10E-05	1,40E-05	2,80E-05	3,20E-05	3,60E-05
5000	4,00E-06	9,00E-06	1,30E-05	1,60E-05	1,22E-02	1,18E-02	1,18E-02	1,17E-02	4,40E-05	3,60E-05	4,30E-05	4,30E-05	1,40E-05	2,90E-05	3,30E-05	3,60E-05
6000	4,00E-06	9,00E-06	1,30E-05	1,60E-05	1,44E-02	1,44E-02	1,44E-02	1,48E-02	4,30E-05	4,30E-05	3,80E-05	4,50E-05	1,40E-05	2,80E-05	3,30E-05	3,60E-05
8000	4,00E-06	9,00E-06	1,30E-05	1,70E-05	1,97E-02	1,97E-02	1,98E-02	1,95E-02	4,10E-05	4,60E-05	4,60E-05	4,50E-05	1,40E-05	3,10E-05	3,50E-05	3,70E-05

4)



5)

5.1) 12 MB Intel Core i7-9750H 2.6 GHz processor, 512GB SSD, 16GB RAM, Nvidia 4GB GTX 1650 GPU and Windows 10 operating system.

5.2) Theoretically:

The best case of algorithm 1 is when the key is the first element of the array and the complexity is $O(1)$. Average case scenario is when the key is in a position closest to the middle of the array. Worst case scenario is if the key is not existent with the complexity $O(N)$.

The best case of algorithm 2 is when the key is the last element of the array with the complexity $O(1)$. Average case is like how it is in algorithm 1, it has to be in the closest position it can get to the middle. Worst case scenario is when the key is not existing with the complexity $O(N)$.

The best case of algorithm 3 is when the key is in the closest position it can get to middle with the complexity $O(1)$. Worst case is when the binary search finds the key at the end of the array when its narrowed down with the complexity of average and worst case being $O(\log n)$.

The best case of algorithm 4 is when the jump search finds the key at the first visited target with the complexity of $O(1)$. Worst case is when jump search finds the key at the end of the array with the complexity of $O(\sqrt{n})$.

5.3)

For algorithm 1, best case scenario is for case 1 which is the same with my theoretical results.

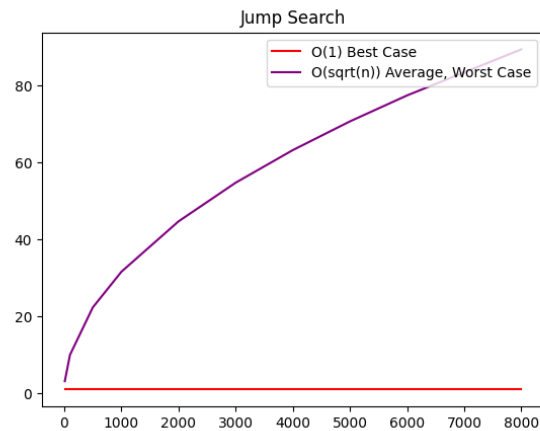
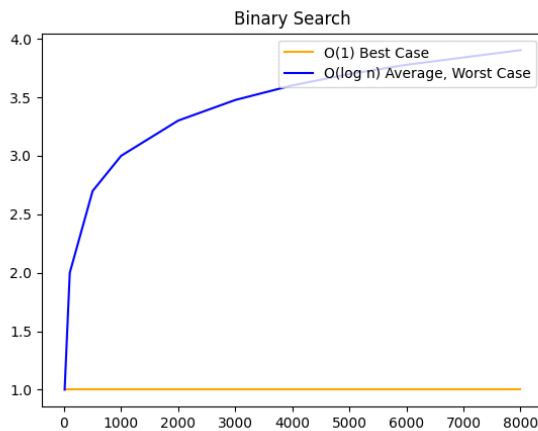
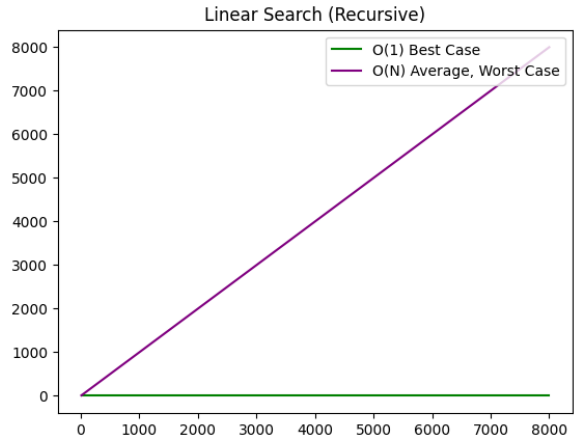
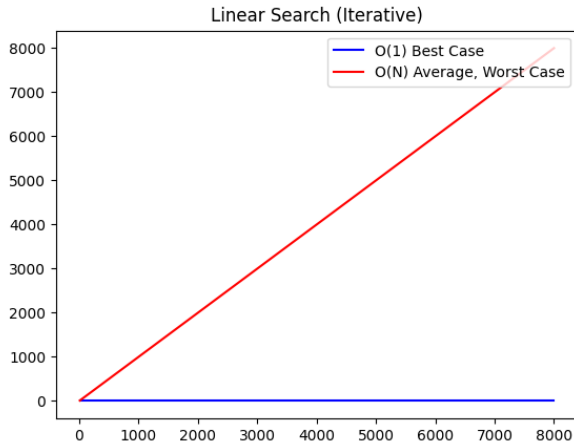
For algorithm 2, according to my observations it is really hard to find a best/worst case scenario since they are almost all the same.

For algorithm 3, it is again hard to find a best/worst case scenario from my observations.

For algorithm 4, best case scenario is for case 1 again which is the same with my theoretical results.

The reasons behind contradictions of theoretical and observed results in algorithms 2 and 3 can be because of several reasons. It can be about not choosing the right indexes for the key values which can result in limited testing field – hard to read graphs. It can be because computer-related issues.

5.4)



Comparing the results I obtained in part 4 and the growth rates I obtained in part 5, it can be said that my results are consistent but they show some differences. The growth rates in part 5 can be sometimes hard to obtain because of the background apps that occupy the computer's memory which slows down the procedure. Which is also why I might get different values every time I run the program.