

EE441 DATA STRUCTURES PROGRAMMING ASSIGNMENT #3**Part 1 – HASHING**

Q1)

Loading Factor:

Folding: 2

Middle Squaring: 2

Truncation: 2

Q2)

Number of Collisions out of 200 entries:

Folding: 114

Middle Squaring: 178

Truncation: 152

Reminder: Last 2 digits of the TC ID is generated according to official guidance,

Therefore; results may differ from completely random TC ID Generation.

Q3)

For this application the Folding has the best performance and Middle Squaring has the worst performance in terms of loading factor.

Part 2 – SORTING

100	#Comparisons	#Moves	Time Elapsed
BubbleSort	4801	2494	0.0772277
SelectionSort	4950	99	0.0277228
QuickSort1	699	160	0.00796813
QuickSort2	663	230	0.0103586
QuickSort3	647	230	0.0111554
QuickSort4	723	230	0.0191235
CustomSort2	589	234	0.00717131

1000	#Comparisons	#Moves	Time Elapsed
BubbleSort	540402	272881	6.63636
SelectionSort	544950	1908	3.8
QuickSort1	17554	3937	0.261538
QuickSort2	17582	5273	0.238462
QuickSort3	17330	5274	0.230769
QuickSort4	18088	5299	0.353846
CustomSort2	15310	5457	0.153846

5000	#Comparisons	#Moves	Time Elapsed
BubbleSort	14447729	7273348	173.4
SelectionSort	14495650	11995	90.4
QuickSort1	146996	31674	1.8
QuickSort2	147291	40808	2.03333
QuickSort3	145852	40801	2.2
QuickSort4	147383	41105	2.56667
CustomSort2	130017	42189	1.7

10000	#Comparisons	#Moves	Time Elapsed
BubbleSort	71630510	35859412	1089.5
SelectionSort	71738475	27991	562.9
QuickSort1	449940	94848	5.13333
QuickSort2	445304	119896	5.66667
QuickSort3	451153	119656	6.06667
QuickSort4	445109	120773	7.26667
CustomSort2	398082	123748	4.86667

25000	#Comparisons	#Moves	Time Elapsed
BubbleSort	>2bil	228201379	6107.2
SelectionSort	>2bil	80982	3011
QuickSort1	1104979	228654	12.6
QuickSort2	1089307	283404	13
QuickSort3	1113033	282384	14.6
QuickSort4	1080798	285727	17.1
CustomSort2	973044	292544	11.8

BubbleSort has the worst characteristics in all 5 cases because it demands maximum number of comparisons and moves. SelectionSort has n^2 characteristic that comes from its high demand in number of comparisons. These make

them not practical when number of elements gets too large.

QS has $n\log(n)$ characteristic in optimistic cases and it can have n^2 at the worst case. QS1 and QS2 has the advantage of not requiring to computation time of rand() function over the QS3 and QS4 that is clearly observable in the Time plot.

The time optimization in CustomSort2 caused the number of comparisons to decrease and number of moves to increase. However, any of the QS based algorithms can perform better than one another when specific data distributions considered, thus implementation must be done according to what is known about the data that is being sorted.



