CPE 202: Data Structures

Fall 2019

Project 3: Sorting Algorithms

Gaurav Joshi and Tushar Sharma

11/10/2019

Professor Toshihiro Kuboi

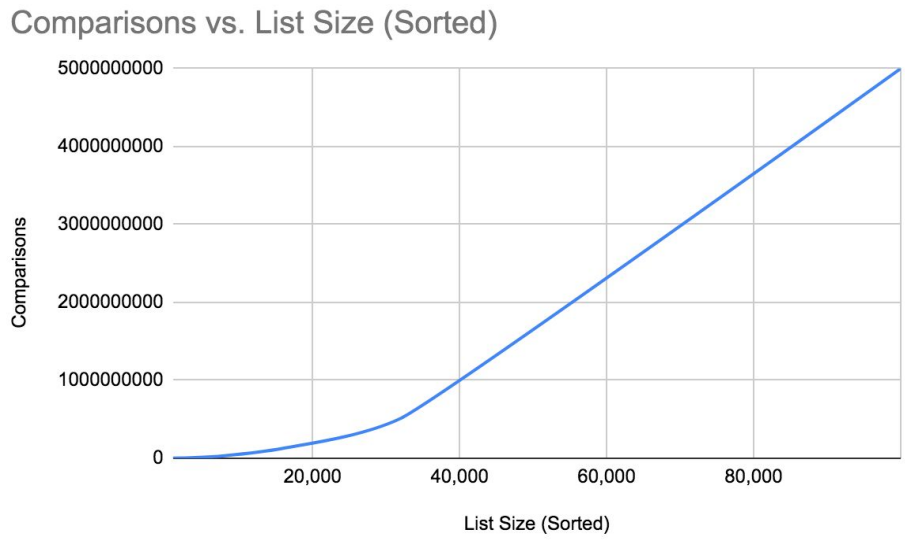
Selection Sort (O(n^2)) - Tushar

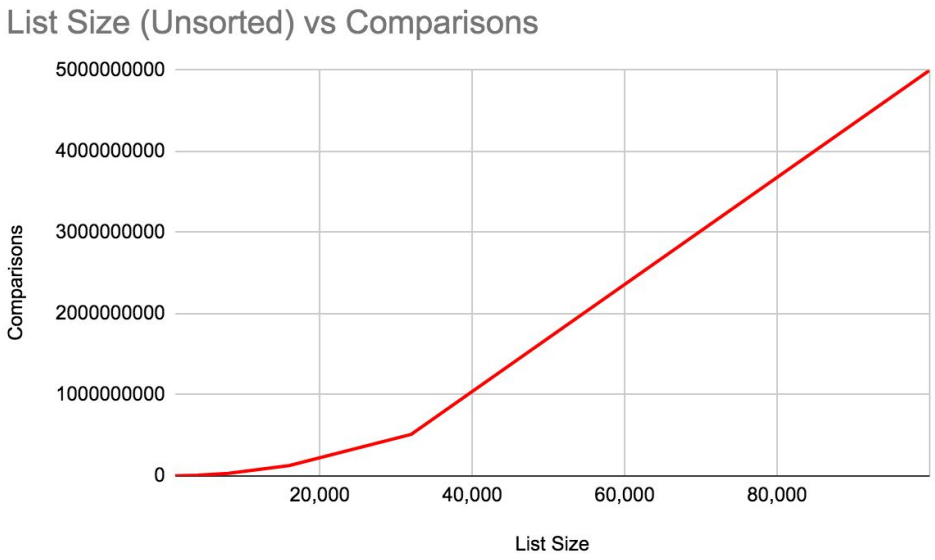
Sorted List:

List Size (Sorted)	Comparisons	Time
1,000(Observed)	499500	0.07662701607
4,000(Observed)	7998000	1.147117853
8,000(Observed)	31996000	4.524917126
16,000(Observed)	127992000	18.76641893
32,000(Observed)	511984000	73.55498886
100,000(Expected)	4999950000	10+ hours
500,000(Expected)	10+hours	10+hours

Unsorted List:

List Size (Unsorted)	Comparisons	Time
1,000 (observed)	499500	0.06887292862
4,000(observed)	7998000	1.09726119
8,000(observed)	31996000	4.736567974
16,000(observed)	127992000	17.87195373
32,000(observed)	511984000	83.14192605
100,000(observed)	4999950000	10+ hours
500,000(observed)	10 +hours	10+ hours





Insertion Sort (O(n^2)) - Gaurav

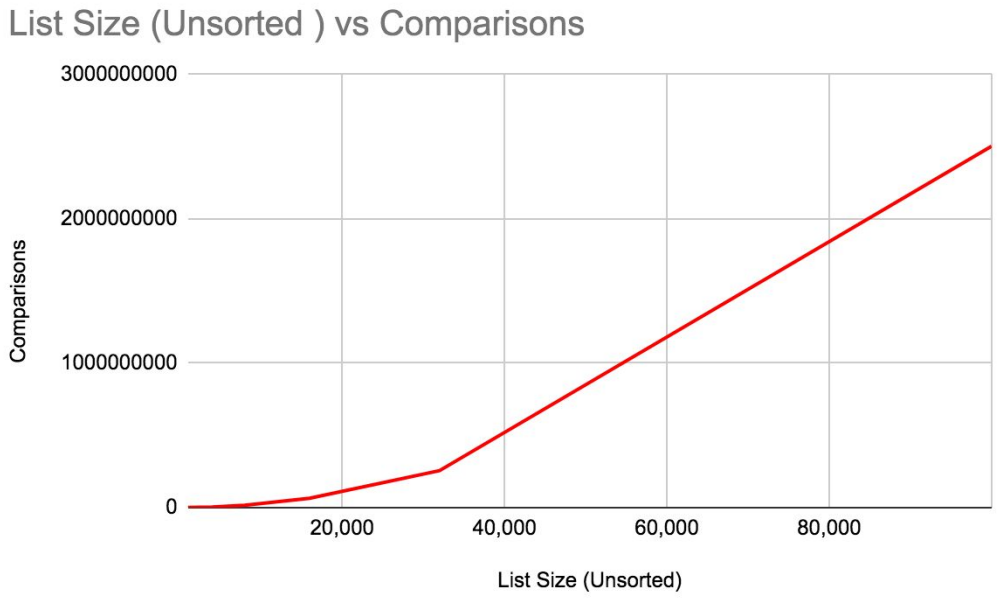
Sorted List:

List Size (Sorted)	Comparisons	Time
1,000(Observed)	999	0.0002250671387
4,000(Observed)	3999	0.0008640289307
8,000(Observed)	7999	0.001885890961
16,000(Observed)	15999	0.007306098938
32,000(Observed)	31999	0.008866071701
100,000(Observed)	99999	0.02235913277
500,000(Observed)	499999	0.148079872131347668

Unsorted List:

List Size (UnSorted)	Comparisons	Time
1,000(Observed)	256233	0.1772217751
4,000(Observed)	4036585	1.83206892
8,000(Observed)	16127455	7.924745083
16,000(Observed)	64259773	30.72982717
32,000(Observed)	256230099	119.4389799

100,000(Observed)	2503327823	10+ hours
500,000(Observed)	10+hours	10+hours



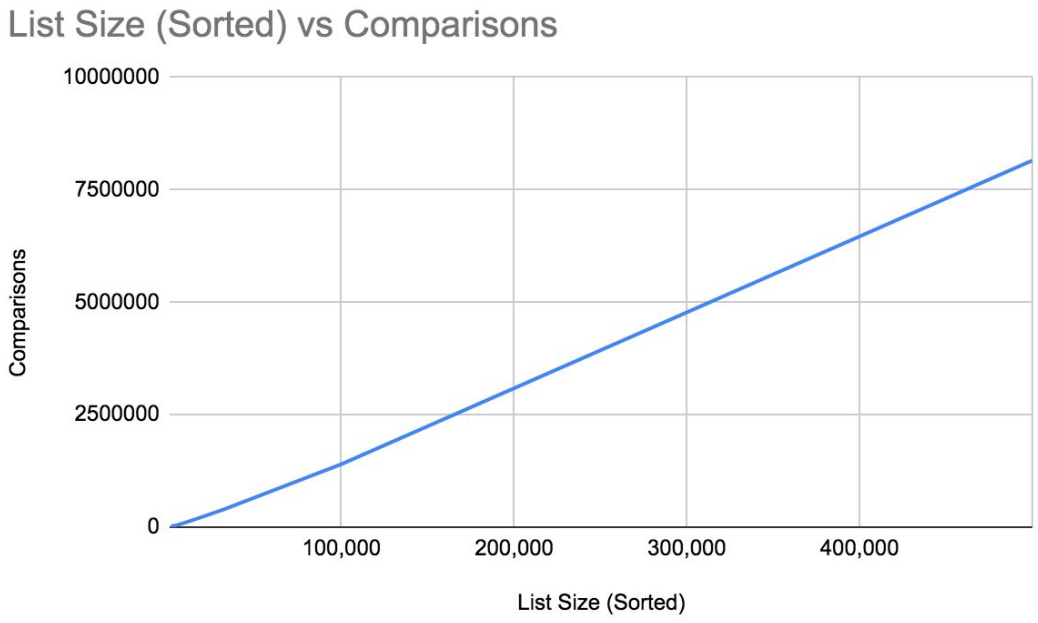
Heap Sort (O(n * log (n))) - Tushar

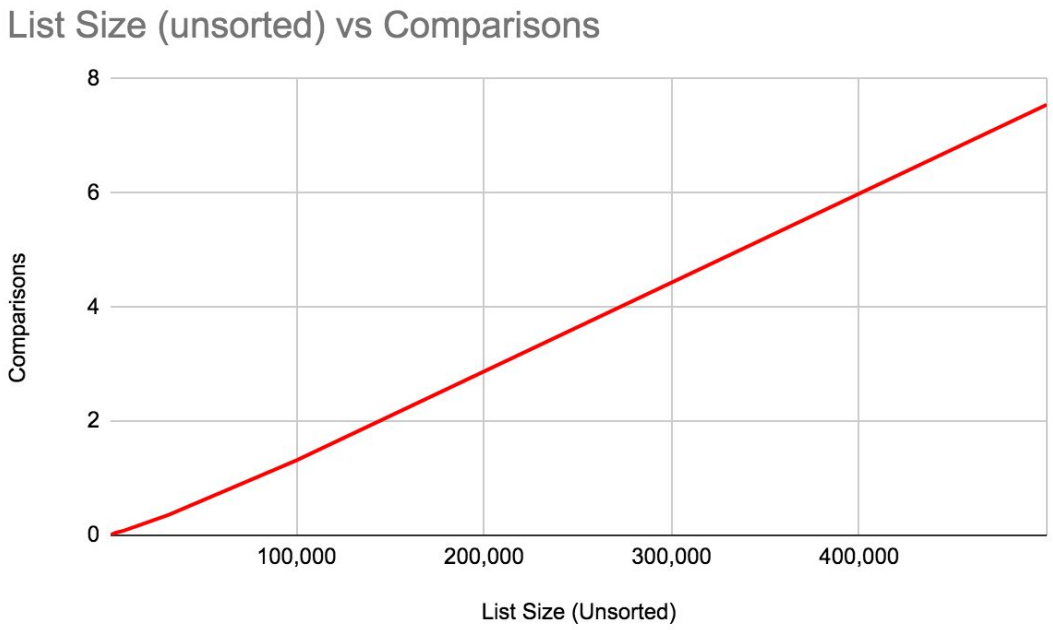
List size sorted:

List Size (Sorted)	Comparisons	Time
1,000(Observed)	89723	0.0101768970489501954
4,000(Observed)	439003	0.044923067092895514
8,000(Observed)	95801	0.100414991378784185
16,000(Observed)	207601	0.20761275291442875
32,000(Observed)	447216	0.44302487373352057
100,000(Observed)	1568908	1.551919937133789
500,000(Observed)	89757074	8.78948974609375

List Size unsorted:

List Size (Unsorted)	Comparisons	Time
1,000(Observed)	8963	0.0131990909576416027
4,000(Observed)	43889	0.04557490348815918
8,000(Observed)	95792	0.09778904914855957
16,000(Observed)	207599	0.20306706428527832
32,000(Observed)	447214	0.45887303352355957
100,000(Observed)	1568908	1.58027029037475597
500,000(Observed)	8975689	8.7656910419464112





Bubble Sort (O(n^2)) - Gaurav

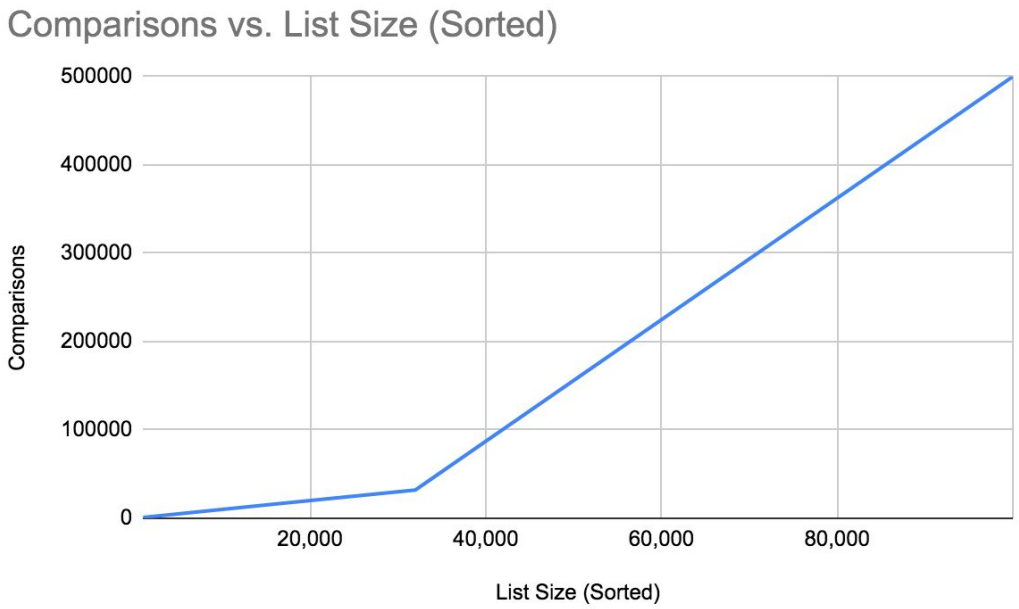
Sorted List:

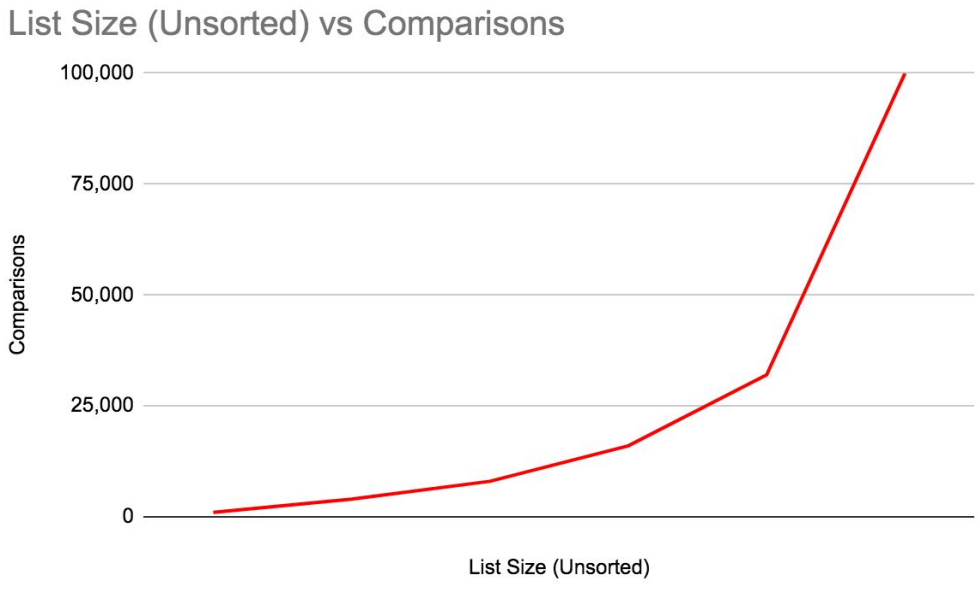
List Size (Sorted)	Comparisons	Time
1,000(Observed)	999	0.07084369659423828
4,000(Observed)	3999	1.1160480976104736
8,000(Observed)	7999	4.459099054336548
16,000(Observed)	15999	20.07073736190796
32,000(Observed)	31999	74.86224102973938
100,000(Expected)	499999	118
500,000(Expected)	10+hours	10+hours

Unsorted List:

List Size (unsorted)	Comparisons	Time

1,000(Observed)	251928	0.15479111671447754
4,000(Observed)	4031708	2.451531171798706
8,000(Observed)	15970220	9.816545009613037
16,000(Observed)	64535040	38.97900700569153
32,000(Observed)	256315568	155.53220677375793
100,000(Expected)	4999846260	10+ hours
500,000(Expected)	10+hours	10+hours

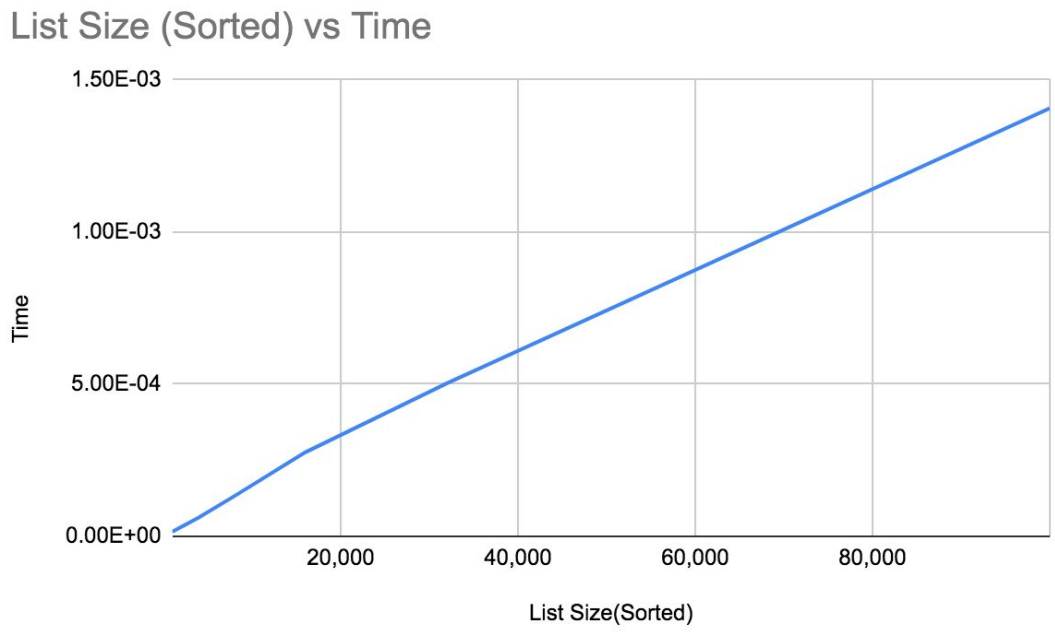


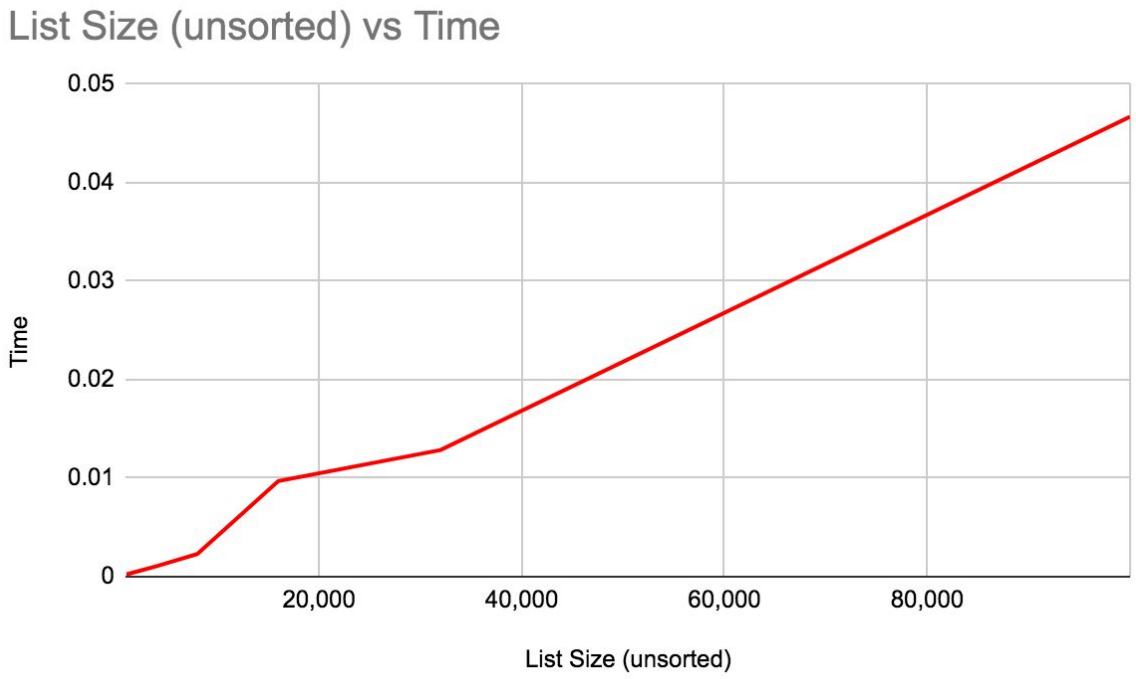


Tim Sort (O(n)) - Tushar

List Size (Sorted)	Comparisons	Comparisons
1,000	N/A	1.48E-05
4,000	N/A	6.20E-05
8,000	N/A	0.0001318454742
16,000	N/A	0.000275850296
32,000	N/A	0.0005030632019
100,000	N/A	0.001406908035

List Size (Sorted)	Comparisons	Time
1,000	N/A	0.0002119541168
4,000	N/A	0.001063108444
8,000	N/A	0.002290010452
16,000	N/A	0.009706020355
32,000	N/A	0.01285099983
100,000	N/A	0.04670500755

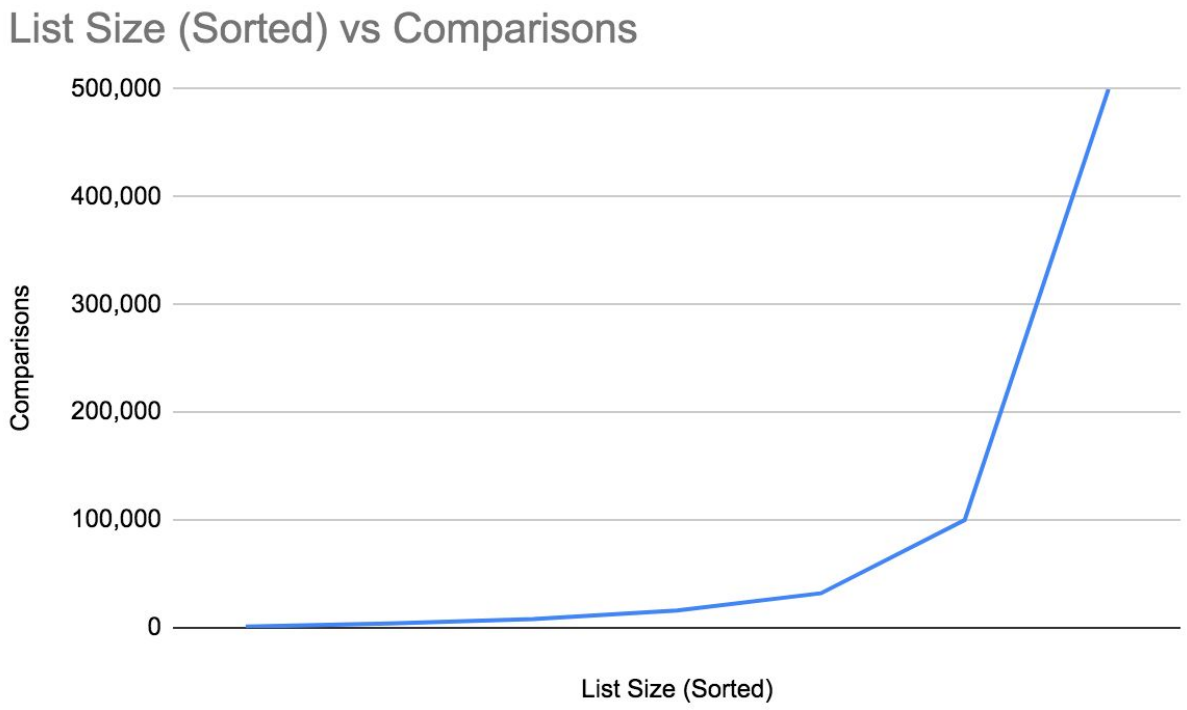


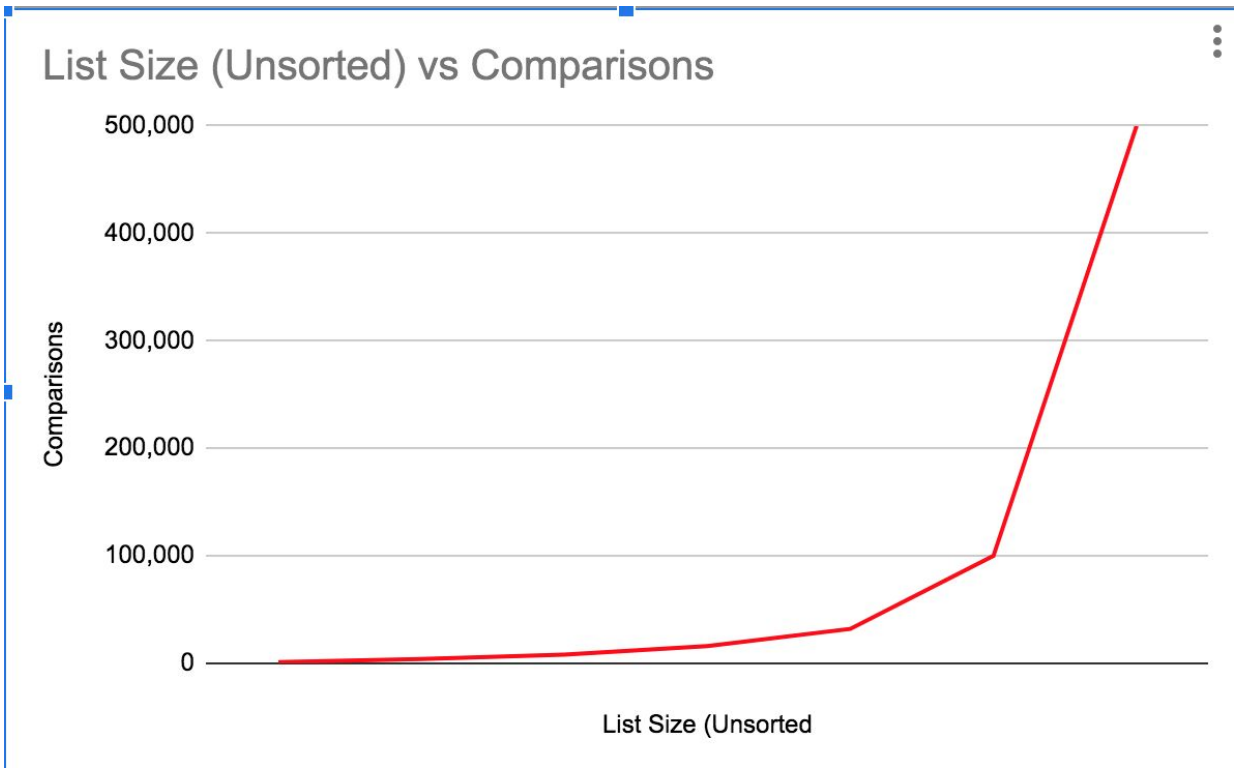


Quick Sort (O(nlogn))- Gaurav

List Size (Unsorted)	Comparisons	Time
1,000	9817	0.005864381790161133
4,000	51261	0.029027700424194336
8,000	114162	0.05896496772766113
16,000	244941	0.11371684074401855
32,000	467857	0.27068090438842773
100,000	1669554	0.8772721290588379
500,000	9362126	4.949580192565918

List Size (Sorted)	Comparisons	Time
1,000	8633	0.006368160247802734
4,000	35943	0.026272058486938477
8,000	78062	0.06044292449951172
16,000	182694	0.11740493774414062
32,000	362282	0.24166297912597656
100,000	1291682	0.788607120513916
500,000	7314636	4.374124050140381




List Size (Unsorted) vs Comparisons
500,000
400,000
300,000
200,000
100,000
0
Comparisons
List Size (Unsorted

Questions

1. Which sort do you think is better? Why?

We believe that outside of tim sort, quick sort is the better of all of the rest of the sorting algorithms. This is for a variety of reasons. First off, it has the best performance in terms of the average case. The best case of quick sort is O(n log n), while the worst case is O(n^2). Moreover, it is O(n log n) in the average case. While it always depends on the input of the unsorted or sorted array, most of the time, the best time complexity can be achieved with quick sort.

2. Which sort is better when sorting a list that is already sorted (or mostly sorted)? Why?

We believe that outside of the tim sort, insertion sort is better when sorting a list that has already been or mostly has been sorted. The reason this occurs is because when the insertion sort is inserting elements in the sorted portion of the array, it barely has to move any elements at all. Thus an insertion sort on a completely sorted array occurs in linear time because all it really needs to do is one comparison per element.

3. You probably found that insertion sort had about half as many comparisons as selection sort. Why? Why are the times for insertion sort not half what they are for selection sort?

With selection sort, there are twice as many comparisons compared to insertion sort as selection sort compares each item to every other item instead of only items in sorted list. Selection Sort has a time complexity of (O (n^2)) in all cases while insertion sorts worst case is (O(n^2)) and best case is O(n).