

Insertion Sort – Total number of comparisons 7

INDEX	0	1	2	3	4	5	6	7	8
UNSORTED	54	25	93	17	77	31	44	55	19
Compare [1st pass]	54	25	93	17	77	31	44	55	19
Swap	54	54	93	17	77	31	44	55	19
Swap	25	54	93	17	77	31	44	55	19
Compare	25	54	93	17	77	31	44	55	19
Swap	25	54	93	93	77	31	44	55	19
Swap	25	54	54	93	77	31	44	55	19
Swap	25	25	54	93	77	31	44	55	19
Swap	17	25	54	93	77	31	44	55	19
Compare	17	25	54	93	77	31	44	55	19
Swap	17	25	54	93	93	31	44	55	19
Swap	17	25	54	77	93	31	44	55	19
Compare	17	25	54	77	93	31	44	55	19
Swap	17	25	54	77	93	93	44	55	19
Swap	17	25	54	77	77	93	44	55	19
Swap	17	25	54	54	77	93	44	55	19
Swap	17	25	31	54	77	93	44	55	19
Compare	17	25	31	54	77	93	44	55	19
Swap	17	25	31	54	77	93	93	55	19
Swap	17	25	31	54	77	77	93	55	19
Swap	17	25	31	54	54	77	93	55	19
Swap	17	25	31	44	54	77	93	55	19
Compare	17	25	31	44	54	77	93	55	19
Swap	17	25	31	44	54	77	93	93	19
Swap	17	25	31	44	54	77	77	93	19
Swap	17	25	31	44	54	55	77	93	19
Compare	17	25	31	44	54	55	77	93	19
Swap	17	25	31	44	54	55	77	93	93
Swap	17	25	31	44	54	55	77	77	93
Swap	17	25	31	44	54	55	55	77	93
Swap	17	25	31	44	54	54	55	77	93
Swap	17	25	31	44	44	54	55	77	93
Swap	17	25	31	31	44	54	55	77	93
Swap	17	25	25	31	44	54	55	77	93
Swap	17	19	25	31	44	54	55	77	93
SORTED	17	19	25	31	44	54	55	77	93

Selection Sort – Total number of comparisons 6

INDEX	0	1	2	3	4	5	6	7	8
UNSORTED	54	25	93	17	77	31	44	55	19
Compare	54	25	93	17	77	31	44	55	19
Swap	54	25	19	17	77	31	44	55	19
Swap	54	25	19	17	77	31	44	55	93
Compare	54	25	19	17	77	31	44	55	93
Swap	54	25	19	17	55	31	44	55	93
Swap	54	25	19	17	55	31	44	77	93
Compare	54	25	19	17	55	31	44	77	93
Swap	54	25	19	17	44	31	44	77	93
Swap	54	25	19	17	44	31	55	77	93
Compare	54	25	19	17	44	31	55	77	93
Swap	31	25	19	17	44	31	55	77	93
Swap	31	25	19	17	44	54	55	77	93
Compare	31	25	19	17	44	54	55	77	93
Swap	17	25	19	17	44	54	55	77	93
Swap	17	25	19	31	44	54	55	77	93
Compare	17	25	19	31	44	54	55	77	93
Swap	17	19	19	31	44	54	55	77	93
Swap	17	19	25	31	44	54	55	77	93
SORTED	17	19	25	31	44	54	55	77	93

Bubble Sort – Total number of comparisons 20

INDEX	0	1	2	3	4	5	6	7	8
UNSORTED	54	25	93	17	77	31	44	55	19
Compare [1st pass]	54	25	93	17	77	31	44	55	19
Swap	25	25	93	17	77	31	44	55	19
Swap	25	54	93	17	77	31	44	55	19
Compare	25	54	93	17	77	31	44	55	19
Swap	25	54	17	17	77	31	44	55	19
Swap	25	54	17	93	77	31	44	55	19
Compare	25	54	17	93	77	31	44	55	19
Swap	25	54	17	77	77	31	44	55	19
Swap	25	54	17	77	93	31	44	55	19
Compare	25	54	17	77	93	31	44	55	19
Swap	25	54	17	77	31	31	44	55	19
Swap	25	54	17	77	31	93	44	55	19

Compare	25	54	17	77	31	93	44	55	19
Swap	25	54	17	77	31	44	44	55	19
Swap	25	54	17	77	31	44	93	55	19
Compare	25	54	17	77	31	44	93	55	19
Swap	25	54	17	77	31	44	55	55	19
Swap	25	54	17	77	31	44	55	93	19
Compare	25	54	17	77	31	44	55	93	19
Swap	25	54	17	77	31	44	55	19	19
Swap	25	54	17	77	31	44	55	19	93
Compare [2nd pass]	25	54	17	77	31	44	55	19	93
Swap	25	17	17	77	31	44	55	19	93
Swap	25	17	54	77	31	44	55	19	93
Compare	25	17	54	77	31	44	55	19	93
Swap	25	17	54	31	31	44	55	19	93
Swap	25	17	54	31	77	44	55	19	93
Compare	25	17	54	31	77	44	55	19	93
Swap	25	17	54	31	44	44	55	19	93
Swap	25	17	54	31	44	77	55	19	93
Compare	25	17	54	31	44	77	55	19	93
Swap	25	17	54	31	44	55	55	19	93
Swap	25	17	54	31	44	55	77	19	93
Compare	25	17	54	31	44	55	77	19	93
Swap	25	17	54	31	44	55	19	19	93
Swap	25	17	54	31	44	55	19	77	93
Compare [3^d pass]	25	17	54	31	44	55	19	77	93
Swap	17	17	54	31	44	55	19	77	93
Swap	17	25	54	31	44	55	19	77	93
Compare	17	25	54	31	44	55	19	77	93
Swap	17	25	31	31	44	55	19	77	93
Swap	17	25	31	54	44	55	19	77	93
Compare	17	25	31	54	44	55	19	77	93
Swap	17	25	31	44	44	55	19	77	93
Swap	17	25	31	44	54	55	19	77	93
Compare	17	25	31	44	54	55	19	77	93
Swap	17	25	31	44	54	19	19	77	93
Swap	17	25	31	44	54	19	55	77	93
Compare [4th pass]	17	25	31	44	54	19	55	77	93
Swap	17	25	31	44	19	19	55	77	93
Swap	17	25	31	44	19	54	55	77	93
Compare	17	25	31	44	19	54	55	77	93
Swap	17	25	31	19	19	54	55	77	93
Swap	17	25	31	19	44	54	55	77	93
Compare	17	25	31	19	44	54	55	77	93
Swap	17	25	19	19	44	54	55	77	93
Swap	17	25	19	31	44	54	55	77	93
Compare	17	25	19	31	44	54	55	77	93
Swap	17	19	19	31	44	54	55	77	93

Swap
SORTED

17	19	25	31	44	54	55	77	93
17	19	25	31	44	54	55	77	93

Quicksort – Total number of comparisons 7

INDEX	0	1	2	3	4	5	6	7	8
<u>UNSORTED</u>	54	25	93	17	77	31	44	55	19
Compare	54	25	93	17	77	31	44	55	19
Swap	54	25	17	17	77	31	44	55	19
	54	25	17	93	77	31	44	55	19
Compare	54	25	17	93	77	31	44	55	19
Swap	54	25	17	31	77	31	44	55	19
	54	25	17	31	77	93	44	55	19
Compare	54	25	17	31	77	93	44	55	19
Swap	54	25	17	31	44	93	44	55	19
	54	25	17	31	44	93	77	55	19
Compare	54	25	17	31	44	93	77	55	19
Swap	54	25	17	31	44	19	77	55	19
	54	25	17	31	44	19	77	55	93
Compare	54	25	17	31	44	19	77	55	93
Swap	19	25	17	31	44	19	77	55	93
	19	25	17	31	44	54	77	55	93
Compare	19	25	17	31	44	54	77	55	93
Swap	19	17	17	31	44	54	77	55	93
	19	17	25	31	44	54	77	55	93
Compare	19	17	25	31	44	54	77	55	93
Swap	17	17	25	31	44	54	77	55	93
	17	19	25	31	44	54	77	55	93
Compare	17	19	25	31	44	54	77	55	93
Swap	17	19	25	31	44	54	55	55	93
	17	19	25	31	44	54	55	77	93
<u>SORTED</u>	17	19	25	31	44	54	55	77	93

Original order:

54 25 93 17 77 31 44 55 19

[54, 25, 93, 17, 77, 31, 44, 55, 19]

Eoll: 5

Pivot Point 54

[19, 25, 17, 31, 44]

Eoll: 1

Pivot Point 19

[25, 31, 44]

Eoll: 2

Pivot Point 25

[31, 44]

Eoll: 3

Pivot Point 31

[77, 55, 93]

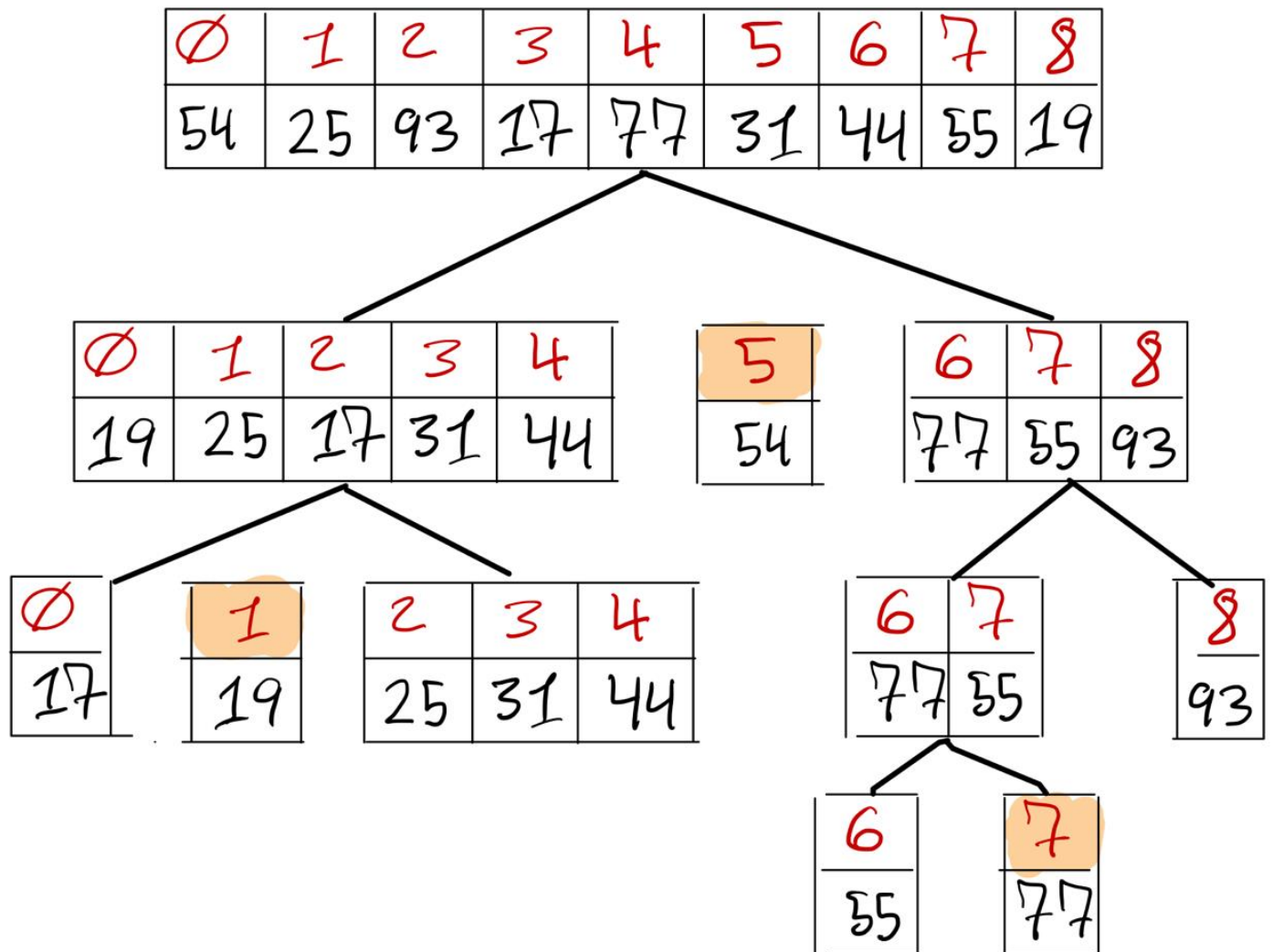
Eoll: 7

Pivot Point 77

After sorting:

17 19 25 31 44 54 55 77 93

Quick Sort trace



0	1	2	3	4	5	6	7	8
54	25	93	17	77	31	44	55	19

#1

0	1	2	3	4	5	6	7	8
54	25	93	17	77	31	44	55	19

Start: 0 End: 8
Point value
EOLL

#2

0	1	2	3	4	5	6	7	8
54	25	93	17	77	31	44	55	19

EOLL
Point value
Scan

#3

0	1	2	3	4	5	6	7	8
54	25	93	17	77	31	44	55	19

Point value
Scan
EOLL

#4

0	1	2	3	4	5	6	7	8
54	25	93	17	77	31	44	55	19

Point value
Scan
swap

#5

0	1	2	3	4	5	6	7	8
54	25	93	17	77	31	44	55	19

Point value
EOLL
swap

#6

0	1	2	3	4	5	6	7	8
54	25	17	93	77	31	44	55	19

Point value
Scan
EOLL

#7

0	1	2	3	4	5	6	7	8
54	25	17	93	77	31	44	55	19

Point value
Scan
swap

#8

0	1	2	3	4	5	6	7	8
54	25	17	93	77	31	44	55	19

Point value
EOLL
for (Scan = Start+1, Scan ≤ End, Scan++)
if (Scan < Point)
End Of Left List ++
Swap (Array, EOLL, Scan)
temp = Array[Scan]
Array[EOLL] = Array[Scan]
Array[Scan] = temp

#9

0	1	2	3	4	5	6	7	8
54	25	17	31	77	93	44	55	19

Point value
Scan
swap

#10

0	1	2	3	4	5	6	7	8
54	25	17	31	77	93	44	55	19

Point value
EOLL
for (Scan = Start+1, Scan ≤ End, Scan++)
if (Scan < Point)
End Of Left List ++
Swap (Array, EOLL, Scan)
temp = Array[Scan]
Array[EOLL] = Array[Scan]
Array[Scan] = temp

#11

0	1	2	3	4	5	6	7	8
54	25	17	31	44	93	77	55	19

Point value
Scan
EOLL

#12

0	1	2	3	4	5	6	7	8
54	25	17	31	44	93	77	55	19

Point value
Scan
swap

#13

0	1	2	3	4	5	6	7	8
54	25	17	31	44	93	77	55	19

Point value
EOLL
for (Scan = Start+1, Scan ≤ End, Scan++)
if (Scan < Point)
End Of Left List ++
Swap (Array, EOLL, Scan)
temp = Array[Scan]
Array[EOLL] = Array[Scan]
Array[Scan] = temp

#14

0	1	2	3	4	5	6	7	8
54	25	17	31	44	19	77	55	93

Point value
Scan
EOLL

#15

0	1	2	3	4	5	6	7	8
54	25	17	31	44	19	77	55	93

Point value
Scan
Scan = End
Swap (Array, Start, EOLL)

#16 start: 0 End: 4

0	1	2	3	4	5	6	7	8
19	25	17	31	44	54	77	55	93

Point value
EOLL
Scan

#17

0	1	2	3	4	5	6	7	8
19	25	17	31	44	54	77	55	93

Point value
EOLL
Scan

#18

0	1	2	3	4	5	6	7	8
19	25	17	31	44	54	77	55	93

Point value
EOLL
Scan

#19

0	1	2	3	4	5	6	7	8
19	25	17	31	44	54	77	55	93

Point value
EOLL
Scan
swap

#20

0	1	2	3	4	5	6	7	8
19	25	17	31	44	54	77	55	93

Point value
EOLL
Scan

#21

0	1	2	3	4	5	6	7	8
19	17	25	31	44	54	77	55	93

Point value
EOLL
Scan

#22

0	1	2	3	4	5	6	7	8
19	17	25	31	44	54	77	55	93

Point value
EOLL
Scan

#23

0	1	2	3	4	5	6	7	8
19	17	25	31	44	54	77	55	93

Point value
EOLL
Scan

#24

0	1	2	3	4	5	6	7	8
19	17	25	31	44	54	77	55	93

Point value
EOLL
Scan

#25

0	1	2	3	4	5	6	7	8
19	17	25	31	44	54	77	55	93

Point value
EOLL
Scan
Scan = End
Swap (Array, Start, EOLL)

#26

0	1	2	3	4	5	6	7	8
19	17	25	31	44	54	77	55	93

Point value
EOLL
Scan

#26 start: 2 End: 4

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

#27

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

#28 start: 3 End: 4

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

#29 start: 3 End: 4

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

#30 start: 6 End: 8

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

#31 start: 6 End: 8

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan
Scan = End
Swap (Array, Start, EOLL)

#32 start: 6 End: 8

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

#33 start: 6 End: 8

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	77	55	93

Point value
EOLL
Scan

Fully Sorted

0	1	2	3	4	5	6	7	8
17	19	25	31	44	54	55	77	93

2. Using the data provided, which of the above algorithm is more efficient and why? Hints: Please consider counting the number of comparisons made by each algorithm. Do not count the number of passes. Please show your work and motivate your answer.

Of the four sorting algorithms Insertion Sort, Selection Sort, Bubble Sort, and Quick Sort, the sorting algorithm that is most efficient for my data set is Selection Sort. When comparing for time complexity, Big-O Notations, of each sort algorithm against the others one can argue that Quick Sort will always triumph over all others; however, this is not always true, for one understand that the “best” sorting algorithms is determined and is dependent upon one’s situation and data set.

He or she must consider the number of elements, the size of an array, that a sorting algorithm must go through to provide the sorted output; for example, if he or she is sorting an array of three elements one might find that Bubble Sort would be the easiest and most efficient to algorithm to implement. In addition, one must also consider if the given data set is mostly sorted; as a result, he or she might find that Heap Sort will outperform Quick Sort.

With this in mind going executing all four algorithms I expected that Quick Sort would on come out on top, for it has the least number of comparisons in a best case scenario based on its Big-O Notation; rather, to my surprise Selection Sort organized my numbers first for a total of six comparisons, with tying in second place both Insertion Sort and Quick Sort for a total of seven comparisons, and lastly Bubble Sort for a total of 20 comparisons.

Below I have a table of my Selection Sort’s element comparisons marked in blue, swaps marked in red, and their final positioning marked in green. It is apparent that a huge majority of my swapped values respective to their final position in the array trends to the right which is not only convenient, but also efficient, for larger numbers are placed closer to their final sorted position; as a result, fewer comparisons fewer comparisons are required to sort the given data.

Selection Sort – Total number of comparisons 6

INDEX	0	1	2	3	4	5	6	7	8
UNSORTED	54	25	93	17	77	31	44	55	19
Compare	54	25	93	17	77	31	44	55	19
Swap	54	25	19	17	77	31	44	55	19
Swap	54	25	19	17	77	31	44	55	93
Compare	54	25	19	17	77	31	44	55	93
Swap	54	25	19	17	55	31	44	55	93
Swap	54	25	19	17	55	31	44	77	93
Compare	54	25	19	17	55	31	44	77	93
Swap	54	25	19	17	44	31	44	77	93
Swap	54	25	19	17	44	31	55	77	93
Compare	54	25	19	17	44	31	55	77	93
Swap	31	25	19	17	44	31	55	77	93
Swap	31	25	19	17	44	54	55	77	93
Compare	31	25	19	17	44	54	55	77	93
Swap	17	25	19	17	44	54	55	77	93
Swap	17	25	19	31	44	54	55	77	93
Compare	17	25	19	31	44	54	55	77	93
Swap	17	19	19	31	44	54	55	77	93
Swap	17	19	25	31	44	54	55	77	93
SORTED	17	19	25	31	44	54	55	77	93

Selection Sort – Total number of comparisons 6

INDEX	0	1	2	3	4	5	6	7	8
UNSORTED	54	25	93	17	77	31	44	55	19
Compare	54	25	93	17	77	31	44	55	19
Swap	54	25	19	17	77	31	44	55	19
Swap	54	25	19	17	77	31	44	55	93
Compare	54	25	19	17	77	31	44	55	93
Swap	54	25	19	17	55	31	44	55	93
Swap	54	25	19	17	55	31	44	77	93
Compare	54	25	19	17	55	31	44	77	93
Swap	54	25	19	17	44	31	44	77	93
Swap	54	25	19	17	44	31	55	77	93
Compare	54	25	19	17	44	31	55	77	93
Swap	31	25	19	17	44	31	55	77	93
Swap	31	25	19	17	44	54	55	77	93
Compare	31	25	19	17	44	54	55	77	93
Swap	17	25	19	17	44	54	55	77	93
Swap	17	25	19	31	44	54	55	77	93
Compare	17	25	19	31	44	54	55	77	93
Swap	17	19	19	31	44	54	55	77	93
Swap	17	19	25	31	44	54	55	77	93
SORTED	17	19	25	31	44	54	55	77	93