Before addressing the two sorting methods of the different cases of arrays, it is imperative to understand how each sort works:

-Insertion Sort: Starts to traverse the array comparing adjacent cells and then stops at a point where the previous index is greater than the current one (starting at 1). It then will store the smaller variable and copy (or shift) all greater cells ahead one index and store the smaller cell behind the shifted cells and then continues indexing. It goes through the whole array inserting each cell into its right position. If the current cell is greater than the previous cell it simply moves to the next one.

-Selection Sort: This sort indexes the whole array and finds the cell with the smallest value. It then stores this value and inserts it at the beginning index of the array and swaps the beginning index to the smallest value's position. The beginning index increases by one index with each full traversal (because previous values would already be sorted smallest to largest).

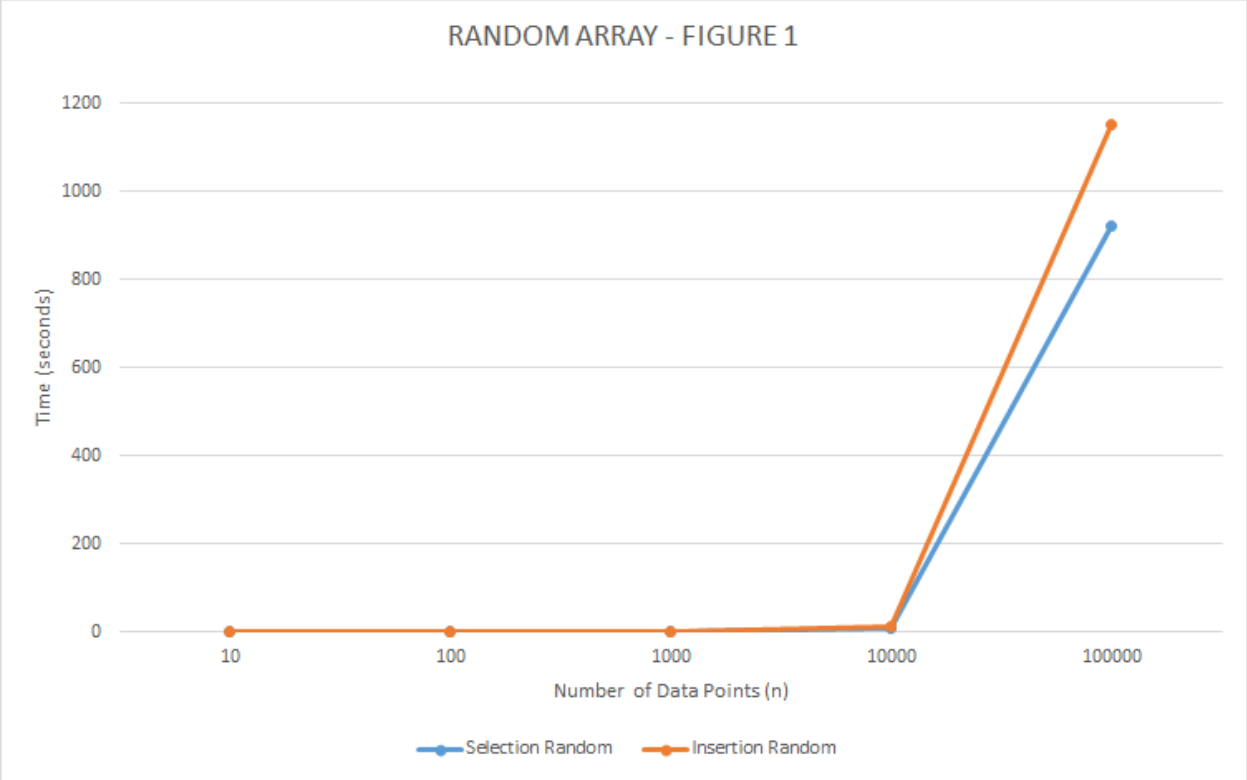
Both the Selection and Insertion sort created sorted subarrays on the left hand side of the actual array. However, Insert Sort computes more time to go through the array as it is sorting because the function is moving values to the left, with a nested while loop conditional that checks to see if the value to the left of the value, *J*, which you are trying to place, is less than *J*, thus the function is going to have more run time, as the number of stored values, *n*, increases unless run through special cases such as an already sorted list. Selection sort negates this problem, using a nested for loop to find the smallest values in the function through the list and sorts them in order, whereas the Insertion sort checks each value over and over to confirm its Boolean while loop expression, the Selection sort has already placed the values in the sorted portion in the correct order, thus it will run at a relatively similar time no matter the position of the contents in the array , while the runtime of the Insertion sort will vary depending on the quantity of contents and their positions relative to each other.

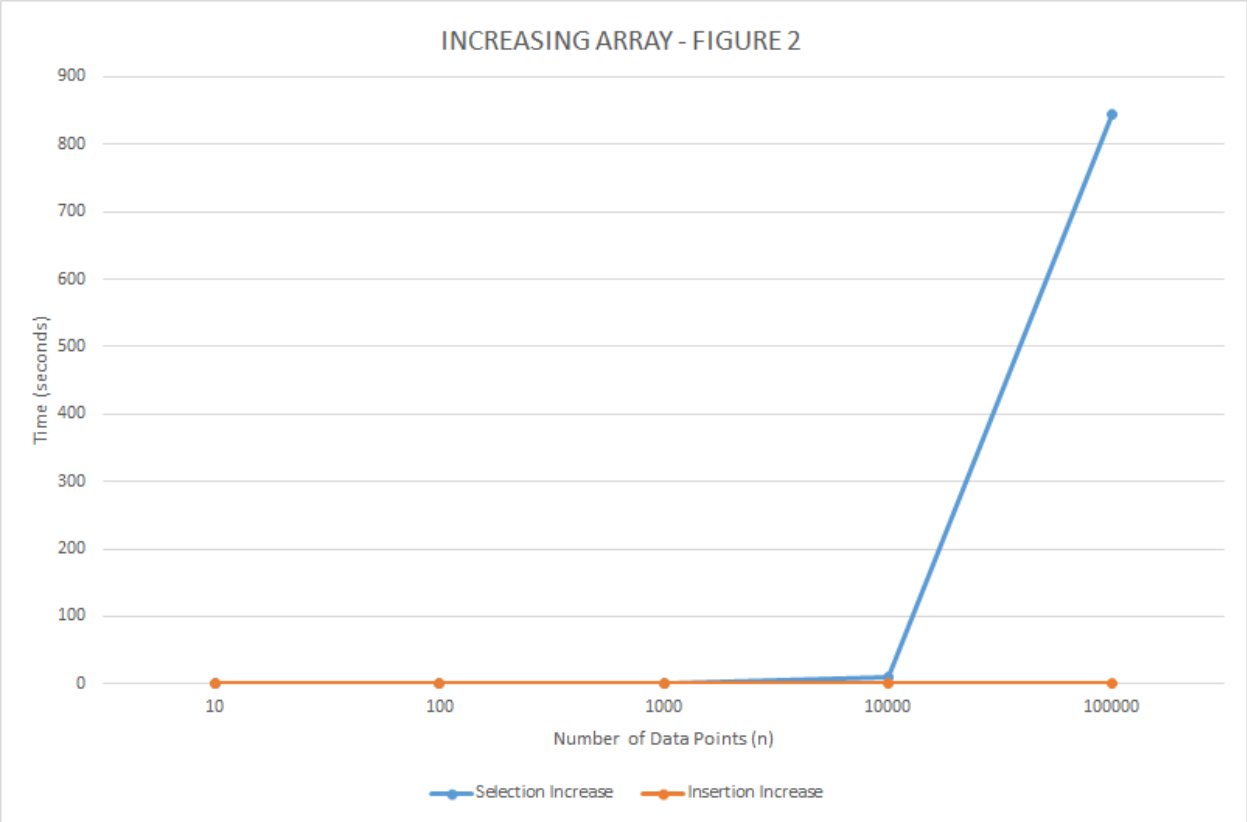
Random Array Sort (Figure 1)
As seen in figure 1, in this case the two sorts were the closest in performance relative to the others. While the difference is least apparent, Insertion Sort took visibly more time than the Selection Sort. This is because when insertion sort is comparing two cells, if the previous is greater than the current it must enter a "for" loop to copy each individual index greater than it ahead one – which is quite a taxing process in terms of efficiency. However, as it will be seen later on in the Selection Sort is not substantially affected by the order of the original array, it simply traverses the array the number of times equal to the length of the array and the act of swapping the of the beginning index and the position of the minimum value is not as taxing as the shift that the insertion sort would entail. When looking at the data, for both sets, when the data points went up by a factor of 10, the time it took to sort went up by around a factor of 100. In this case it gives them both a time complexity of O(n^2) - quadratic time.

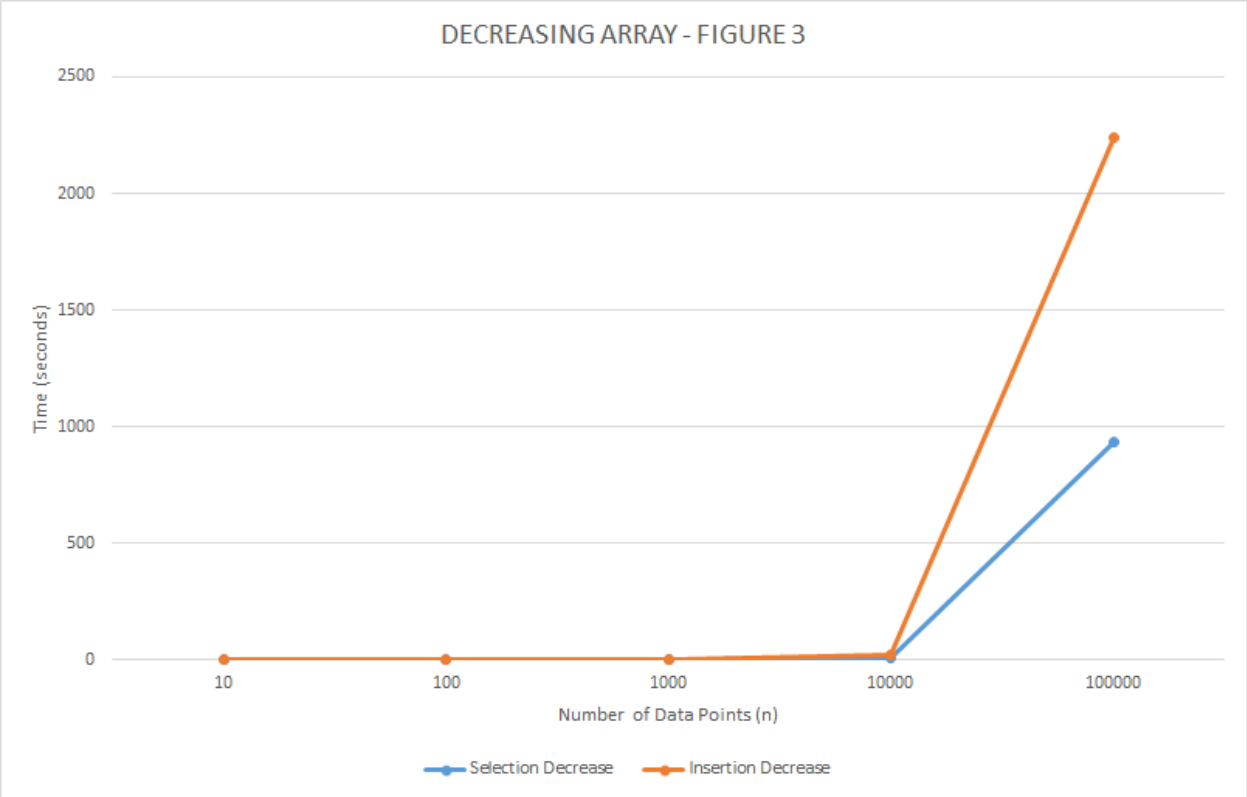
Increase Array Sort (Figure 2)
As seen in figure 2, the Insertion Sort performed substantially better than the selection sort. The selection sort, as expected held around the same time and time complexity as before of O(n^2). However, the given case of sorting an Increasing Array is the best case for an insertion sort. Insertion sort goes through each one of the indexes of the array once, and only performs its shifting function by entering the "for" loop at points at which the two adjacent cells of the array are out of order. This means for an increasing array, it will never have to perform the shift. Selection sort on the other hand still fully indexes the array the number of times equal to the length of the array. When looking at the times for the insertion sort, it can be seen that when the data set increased by a factor of 10, the time increased by a factor of 10 as well. This displays that the best case for the Insertion Sort has a time complexity of O(n) - linear time.

Decrease Array Sort (Figure 3)
Opposite of the previous case, in this example the array is decreasing thus giving the Insertion Sort a substantially higher time than the Selection Sort. The case of the decreasing array is now the worst possible input for the Insertion Sort. Since at every index the sort goes through the two adjacent cells are not in order, it must enter the "for" loop and perform its shift function on every other cell of the array that is greater than the value of the current index. The Selection Sort timing remains the same as it is still traversing the array and equal amount of times to find the minimum and perform the swap of the beginning index and the minimum value index.

Conclusions
What is concluded from this test is that the Selection Sort has a relatively similar time complexity among all cases while Insertion Sorts time complexity varies drastically depending on the case. This makes sense because when thinking about how the sorts work, the Selection Sort will always go through the array an equal amount of times across all scenarios. The Insertion Sort will always go through the array once, but the bulk of the time consumption comes in when it has to enter the "for" loop and perform the shift of the array. The number of times it has to shift the array is completely dependent on the order of the original array.




INCREASING ARRAY - FIGURE 2
900
800
700
600
500
400
300
200
100
0
Time (seconds)
10
100
1000
10000
100000
Number of Data Points (n)
Selection Increase
Insertion Increase


DECREASING ARRAY - FIGURE 3
2500
2000
1500
1000
500
0
Time (seconds)
10
100
1000
10000
100000
Number of Data Points (n)
Selection Decrease
Insertion Decrease

N	Selection Random Time	Insertion Random Time
10	2.22E-05	1.47E-05
100	0.000899	0.000929
1000	0.08831	0.107263
10000	8.813211	10.96228
100000	921.216	1152.351
N	Selection Increase Time	Insertion Increase Time
10	1.88E-05	5.59E-06
100	0.000831	3.36E-05
1000	0.90212	0.000385
10000	8.785775	0.003925
100000	844.9509	0.039493
N	Selection Decrease Time	Insertion Decrease Time
10	2.18E-05	2.09E-05
100	0.000879	0.00173
1000	0.95075	0.190641
10000	9.015989	21.4141
100000	935.0953	2241.553

DATA

1. Selection RANDOM: 10 2.796262562063921e-05
Selection RANDOM: 10 2.0971969215479412e-05
Selection RANDOM: 10 2.068069186526442e-05
Selection RANDOM: 10 2.0389414515049425e-05
Selection RANDOM: 10 2.097196921547941e-05
AVERAGE:

2. Insertion RANDOM: 10 1.9224305114189452e-05
Insertion RANDOM: 10 1.2233648709029653e-05
Insertion RANDOM: 10 1.165109400859968e-05
Insertion RANDOM: 10 1.1651094008599666e-05
Insertion RANDOM: 10 1.8933027763974472e-05
AVERAGE:

3. Selection INCREASE: 10 1.861434057429407e-05
Selection INCREASE: 10 1.864175041375948e-05
Selection INCREASE: 10 1.8641750413759465e-05
Selection INCREASE: 10 1.893302776397446e-05
Selection INCREASE: 10 1.893302776397446e-05
AVERAGE:

4. Insertion INCREASE: 10 6.408101704729799e-06
Insertion INCREASE: 10 5.53426965408486e-06
Insertion INCREASE: 10 5.242992303869853e-06
Insertion INCREASE: 10 5.534269654084833e-06
Insertion INCREASE: 10 5.242992303869853e-06
AVERAGE:

5. Selection DECREASE: 10 2.708879356999426e-05
Selection DECREASE: 10 1.922430511418948e-05
Selection DECREASE: 10 2.4758574768274312e-05
Selection DECREASE: 10 1.922430511418948e-05
Selection DECREASE: 10 1.8933027763974472e-05
AVERAGE:

6. Insertion DECREASE: 10 2.5923684169134285e-05
Insertion DECREASE: 10 1.980685981461944e-05
Insertion DECREASE: 10 2.0098137164834446e-05
Insertion DECREASE: 10 2.009813716483442e-05
Insertion DECREASE: 10 1.8641750413759492e-05
AVERAGE:

7. Selection RANDOM: 100 0.0009405345638442087
Selection RANDOM: 100 0.0007742151968714482
Selection RANDOM: 100 0.0010774349184452547
Selection RANDOM: 100 0.0007969348301882176
Selection RANDOM: 100 0.0007593600520104837
AVERAGE:

8. Insertion RANDOM: 100 0.0008531513587797109
Insertion RANDOM: 100 0.0008717931091934707
Insertion RANDOM: 100 0.0012175393238986658
Insertion RANDOM: 100 0.0007919831152345628
Insertion RANDOM: 100 0.0009122806608733542
AVERAGE:

9. Selection INCREASE: 100 0.0007914005605341328
Selection INCREASE: 100 0.000776836693023383
Selection INCREASE: 100 0.0010555891171791306
Selection INCREASE: 100 0.0007654768763649985
Selection INCREASE: 100 0.0007646030443143538
AVERAGE:

10. Insertion INCREASE: 100 3.262306322407904e-05
Insertion INCREASE: 100 2.9710289721928806e-05
Insertion INCREASE: 100 4.165266108074363e-05
Insertion INCREASE: 100 3.1749231173434315e-05
Insertion INCREASE: 100 3.204050852364951e-05
AVERAGE:

11. Selection DECREASE: 100 0.0008161591353024068
Selection DECREASE: 100 0.0007931482246354227
Selection DECREASE: 100 0.0010523850663267656
Selection DECREASE: 100 0.000865093730138526
Selection DECREASE: 100 0.0008694628903917504
AVERAGE:

12. Insertion DECREASE: 100 0.0016221235633472894
Insertion DECREASE: 100 0.0016873696897954475
Insertion DECREASE: 100 0.002174676696705129
Insertion DECREASE: 100 0.0015670721441566563
Insertion DECREASE: 0.00160115159413181
AVERAGE:

13. Selection RANDOM: 1000 0.09720595114579772
Selection RANDOM: 1000 0.09028869663289209
Selection RANDOM: 1000 0.08755214592762224
Selection RANDOM: 1000 0.08250401817104622
Selection RANDOM: 1000 0.08399914480969978
AVERAGE:

14. Insertion RANDOM: 1000 0.12154741674856415
Insertion RANDOM: 1000 0.1099962308710882
Insertion RANDOM: 1000 0.1056127980277028
Insertion RANDOM: 1000 0.10223718481606126
Insertion RANDOM: 1000 0.09692049934258702
AVERAGE:

15. Selection INCREASE: 1000 0.10135053656200682
Selection INCREASE: 1000 0.09875234259808913
Selection INCREASE: 1000 0.08322959005043173
Selection INCREASE: 1000 0.08530698011216509
Selection INCREASE: 1000 0.08242042157153454
AVERAGE:

16. Insertion INCREASE: 1000 0.0004937151086144564
Insertion INCREASE: 1000 0.0003955546415919664
Insertion INCREASE: 1000 0.00037137862152414325
Insertion INCREASE: 1000 0.0003431247185532449
Insertion INCREASE: 1000 0.0003209876399369316
AVERAGE:

17. Selection DECREASE: 1000 0.11566914854387539
Selection DECREASE: 1000 0.09861340330203655
Selection DECREASE: 1000 0.0890667881487402
Selection DECREASE: 1000 0.08808227070501351
Selection DECREASE: 1000 0.08394525849991002
AVERAGE:

18. Insertion DECREASE: 1000 0.1987024176602622
Insertion DECREASE: 1000 0.1975967288388461
Insertion DECREASE: 1000 0.18485421859899093
Insertion DECREASE: 1000 0.18659518332122588
Insertion DECREASE: 1000 0.18545716271393597
AVERAGE:

19. Selection RANDOM: 10000 9.25630775665758
Selection RANDOM: 10000 8.95157747074556
Selection RANDOM: 10000 8.866955866238449
Selection RANDOM: 10000 8.607470564967374
Selection RANDOM: 10000 8.38374159737664
AVERAGE:

20. Insertion RANDOM: 10000 10.922055346191868
Insertion RANDOM: 10000 10.92177047694336
Insertion RANDOM: 10000 11.315855624303483
Insertion RANDOM: 10000 11.158099228872345
Insertion RANDOM: 10000 10.493610540045683
AVERAGE:

21. Selection INCREASE: 10000 9.140314416422914
Selection INCREASE: 10000 8.726784758271837
Selection INCREASE: 10000 8.500714794617426
Selection INCREASE: 10000 9.062336848274214
Selection INCREASE: 10000 8.498725370315459
AVERAGE:

22. Insertion INCREASE: 10000 0.00403652151927858
Insertion INCREASE: 10000 0.004137012205106316
Insertion INCREASE: 10000 0.003825636717724734
Insertion INCREASE: 10000 0.0039060292663819496
Insertion INCREASE: 10000 0.0037178640981423428
AVERAGE:

23. Selection DECREASE: 10000 9.538517060405677
Selection DECREASE: 10000 8.814091357393227
Selection DECREASE: 10000 8.924455180280294
Selection DECREASE: 10000 8.923490469696379
Selection DECREASE: 10000 8.879388748655028
AVERAGE:

24. Insertion DECREASE: 10000 22.445717553013935I
Insertion DECREASE: 10000 20.58664802103256
Insertion DECREASE: 10000 20.981403688852872
Insertion DECREASE: 10000 21.91668477440861
Insertion DECREASE: 10000 21.140067995784634
AVERAGE:

25. Selection RANDOM: 100000 897.7320254203569
Selection RANDOM: 100000 917.5721272625697
Selection RANDOM: 100000 910.5067890924788
Selection RANDOM: 100000 961.9255538784453
Selection RANDOM: 100000 918.343683388511
AVERAGE:

26. Insertion RANDOM: 100000 1120.1216033419998
Insertion RANDOM: 100000 1189.7625006044004
Insertion RANDOM: 100000 1186.3181310829634
Insertion RANDOM: 100000 1114.9235149369938
Insertion RANDOM: 100000 1150.6293740391486
AVERAGE:

27. Selection INCREASE: 100000 881.7053330552603
Selection INCREASE: 100000 837.5153520640206
Selection INCREASE: 100000 864.2865630263018
Selection INCREASE: 100000 822.160303615859
Selection INCREASE: 100000 819.0870732859639
AVERAGE:

28. Insertion INCREASE: 100000 0.04482641908862206
Insertion INCREASE: 100000 0.03917942510042849
Insertion INCREASE: 100000 0.04034628216504643
Insertion INCREASE: 100000 0.03751885292649604
Insertion INCREASE: 100000 0.03559380091883213
AVERAGE:

29. Selection DECREASE: 100000 947.2009586520148
Selection DECREASE: 100000 948.096735538225
Selection DECREASE: 100000 917.5123353045042
Selection DECREASE: 100000 917.5123353045042
Selection DECREASE: 100000 945.1543691311258
AVERAGE:

30. Insertion DECREASE: 100000 2204.590472783918
Insertion DECREASE: 100000 2376.009894691587
Insertion DECREASE: 100000 2202.4728678061047
Insertion DECREASE: 100000 2212.3453806616308
Insertion DECREASE: 100000 2212.3453806616308
AVERAGE: