|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr** | **Sort** | **Number elements** | **Order** | **Time taken** |
| 1 | Bubble sort | 10 | Ascending | 3.69190092897042e-05 |
| 2 | Bubble sort | -100 | Descending | 5.266199877951294e-05 |
| 3 | Bubble sort | -12345 | Random Order | 4.516198532655835e-05 |
| 4 | Bubble sort | -88 | Nearly sorted | 6.3226500060409331e-05 |
| 5 | Bubble sort | 25 | Random Order | Array not in order |
| 6 | Insertion sort | 100 | Ascending | 9.3014009608887136e-05 |
| 7 | Insertion sort | 123 | Descending | Array not in order |
| 8 | Insertion sort | 2 | Random Order | Array not in order |
| 9 | Insertion sort | 10 | Nearly sorted | Array not in order |
| 10 | Insertion sort | 100000 | Ascending | 0.076641512001049705 |
| 11 | Selection sort | 500 | Ascending | 0.00020556749950628728 |
| 12 | Selection sort | 10 | Descending | Array not in order |
| 13 | Selection sort | 120 | Random order | Array not in order |
| 14 | Selection sort | 10 | Nearly sorted | Array not in order |
| 15 | Selection sort | 1000 | Random order | Array not in order |

In terms of time complexity, the bubble sort does a (N-P) comparison, the best-case scenario for the bubble sort is that all the elements are already in the sorted order, the complexity for this is O(N) which is passing through each element only once, worst-case is that the elements are in a reverse-sorted order or descending order as well as the averages cases fewer swaps, both average and worst case have the complexity of O(N2), it takes a longer time to finish due to swaps and comparisons. Next is the insertion sort, insertion sort finds the right place to insert. The best case is all the elements are in order, it only needs to go through the elements once resulting in the complexity of O(N). The time complexity of O(N2) applies to both the average and worst-case scenarios either in a reverse-sorted order or the elements are scrambled, the worst case for insertion sort is that it must through all the elements until the last one, the average case is that it only needs to go through half the elements each pass. Lastly, the selection sort also does a (N-P) comparison, all best, average, and worst-case scenarios have the time complexity of O(N2) because passes through the list to identify the smallest element, each swap occurs after each pass, the swapped elements are sorted so it does not have to be checked. The characteristics of, the bubble sort is that larger values are always the first to be sorted and takes only one iteration to know that a collection is already sorted. The insertion sort is efficient for smaller data sets whereas inefficient for larger data sets, it is also adaptive. The selection sort is the best option of algorithm when doing swaps for a costly operation, it also a comparison-based sorting algorithm.