Insertion sort is a sorting algorithm that works by selecting an array element and inserting it into its rightful position within a sorted sub-list in the array. It does so by swapping indexes with elements that have greater value than itself. If that is not the case the iterations continue until the array or list is sorted in a particular order. Insertion sort is somewhat similar to bubble sort but is more efficient as it reduces element comparisons with each iteration. Although this algorithm is more efficient than the bubble sort, it is still inefficient compared to many other sort algorithms since it, and a bubble sort, move elements only one position at a time, and due to its inefficiency, cannot be used on very large input size. The best case of scenario of the insertion sort algorithm on input is when the input is already sorted. This gives the best case to be O(n) since the iterations still take place until the end of the list. The average case analysis of a tightly bound algorithm is mathematically achieved to be O(n2). The simplest worst-case input is an array sorted in reverse order. The set of all worst-case inputs consists of all arrays where each element is the smallest or second-smallest of the elements before it. In these cases, every iteration of the inner loop will scan and shift the entire sorted subsection of the array before inserting the next element, making the worst case of the insertion sort algorithm also O(n2). A typical instance of insertion sort being used in real life is when an individual is sorting a deck of cards.