10662016

CSCD 304

Insertion sort is a simple sorting algorithm that build the final sorted array one item at a time. In layman’s terms, the insertion sort algorithm is what is used by card players when shuffling their cards. Insertion sorts iterates and after every iteration, the sort removes one element from the data then looks for the location to which it belongs in the list. It then inserts it there. Insertion sort is one of the easier algorithms to implement and unlike selection sort and bubble sort which are built on the foundation of comparing and swapping, insertion sorts by identifying elements that are out of order relative to the elements around it, removing it from the list, shifting elements up one place and then placing the removed element in its correct location. The best times to use this algorithm are: When data set is relatively small and when the items in the data set are semi-sorted already.

Below are the time complexities of the algorithm:

* Worst-case performance is O(n²) meaning that if you have 100 elements in your array, at worst, the algorithm will do 10,000 comparisons.
* Average case performance: O(n²) — The average case won’t quite be 10,000 comparisons, but it will be higher than O(n) or O(nlogn) time.
* Best case performance: O(n) — The best-case performance is when the data set is already sorted and it only needs to iterate each item in the array and do one comparison for each item in the data set.