Algorithms

[Author Name(s), First M. Last, Omit Titles and Degrees]

[Institutional Affiliation(s)]

Algorithms for Sorting

### The algorithms that have been selected are bubble sort, insertion sort, and selection sort.

Bubble sort compares the adjacent elements to see if they are in the right order i.e. ascending or descending. If they are in the correct order, it moves to the next two elements, otherwise, the positions of the elements are swapped. Each time the full array is traversed, one element moves to the right location after each outer-loop is executed. Its best case with regards to time is O(n) and worst is O(n^2). It involves two loops and each time the first loop is executed, the second loop executes one less time because one element is always sorted. Bubble sort is utilized for a quick implementation regardless of the time constraint demanded (Min, 2010).

Insertion sort is another sorting algorithm. It works by traversing through the list of elements and checking if that particular element is in the right order. If the element is not in the right order, it is placed in the right position and then the algorithm moves onto the next element. The process ends when the list is fully sorted. The best case for the insertion sort is O(n) and the worst is O(n^2). Insertion sort is used when the list size is small as it would cause issues for larger lists (Wang Min, 2010).

The third type of sorting technique is called selection sort. Let us consider a list that is to be sorted. The sorting order is ascending. The selection sort algorithm would take the minimum value element and then place it at the beginning. The process would continue till the entire list is perfectly sorted. When talking about selection sort, the best time complexity is O (n^2) while its worst-case complexity is also O (n^2). Selection sort is also utilized when time is not an issue and easy implementation is demanded. If properly optimized, selection sort may be quicker than the insertion sort (Jadoon et al., 2011).

I would choose bubble sort as it is among the quickest algorithms that have been listed. It has the most efficient best-case time complexity. Furthermore, it is one of the easiest algorithms to implement so it would be my main choice for an algorithm when implementing in the application.

The pseudocode for the implementation of bubble sort is as follows. ‘ar’ is the array of total size ‘p’.

for s =1 to p

for k = 0 to p-1

if a[k]>a[k+1]

swap(ar[k], ar[k+1])

**References**

Jadoon, S., Solehria, S. F., & Qayum, M. (2011). Optimized selection sort algorithm is faster than insertion sort algorithm: a comparative study. *International Journal of Electrical & Computer Sciences IJECS-IJENS*, *11*(02), 19–24.

Min, W. (2010). Analysis on Bubble Sort Algorithm Optimization. *2010 International Forum on Information Technology and Applications*, 208–211. https://doi.org/10.1109/IFITA.2010.9

Wang Min. (2010). Analysis on 2-element insertion sort algorithm. *2010 International Conference On Computer Design and Applications*, V1-143-V1-146. https://doi.org/10.1109/ICCDA.2010.5541165