Comparative Study of Various Enhanced Selection Sort Algorithm

Dr. K. Kavitha
Assistant Professor
Department of Computer Science
Mother Teresa Women's University
Kodaikanal.

***Abstract--* Sorting is commonly used process in data structures to arrange the data in a specific order for efficient element search. Various ways can be used to list the items in order. Many algorithms have been developed still, the researchers proposed many ideas for arranging the data in proper order. Selecting Sort is the simplest algorithm which improves the performance better than Bubble Sort. Enhanced Selection Sort algorithms are introduced by many researchers. This paper discussed about Classical selection Sort with Enhanced Selection Sort Algorithm and analyzes the benefits and limitations of each algorithm. The main objective of this study is to point out the best selection sort algorithm based on the performance analysis.**

Keywords:- Sort, Selection Sort, ESS, ARC, ESSA, Swap

I. INTRODUCTION

Sorting process is for data arrangement in a specific order which locates the information in an efficient way. An array of integers may be sorted from lower to higher or from higher to lower (or) array of string may be sorted either in ascending or in descending. Most of the sorting algorithms like selection sort, bubble sort, merge sort etc uses swapping element technique till reach the goals.

Each algorithm is chosen based on best case, average and worst case. Accordingly, selection sort is the best one to arrange the elements. Many researchers proposed enhanced selection sort algorithms for avoiding unnecessary comparisons. This paper presents the comparative analysis of the classical selection sort and enhanced selection sort algorithms proposed by various authors.

This paper is organized as follows

Section-2 gives the brief explanation of the selection sort algorithm

Section3 illustrates the working procedure, benefits and limitations of enhanced selection sort algorithms

Section-4 concludes the review and

Section-5 gives important reference.

II. CLASSICAL SELECTION SORT ALGORITHM

It identifies the maximum value in the list and swaps the last element within it. Then it looks for the next maximum value in the entire list. This process starts from the beginning of the list and let it be the maximum element. Continues the process till to identify the maximum value in the entire list.

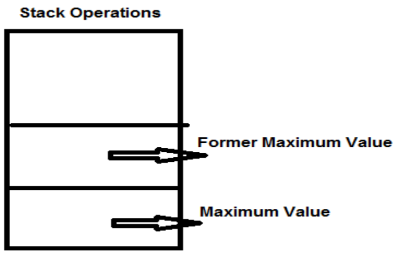
A. Limitations

- Maximize the Search Space
- Performs O(n) Swaps
- Unnecessary Comparisons
- Running Time is too High

III. STUDY OF ENHANCED SELECTION SORT ALGORITHMS

A.Enhanced Selection Sort Algorithm [ESSA]

At the same time, Maximum and Former Maximum value can be identified by Stack. Previous maximum value can be tracked when a new maximum value is identified and it can be used later. And then the next iteration starts from the previous maximum value which guarantees that no other value is greater than former value. It avoids unnecessary Comparisons and minimizes the search space comparable with classical algorithm [1]



Benefits

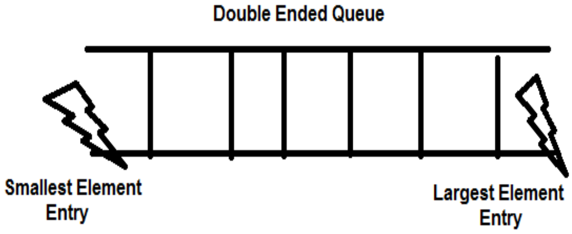
- Multiple Maximum values can be identified in a single iteration
- Stack operations are applied
- Unnecessary Swapping is avoided
- Minimizing the Comparisons

Limitations

- Keeps extra memory space to maintain Stack Structure.

B.Modified Double Ended Selection Sort

It bulges out from two elements, search the entire list and key out the minimum and maximum value. Minimum value stores in the first place and Maximum value stores in the last blank space then select next minimum and maximum in the last excluding first and last element. This process continues till the list is sorted [2]



Benefits

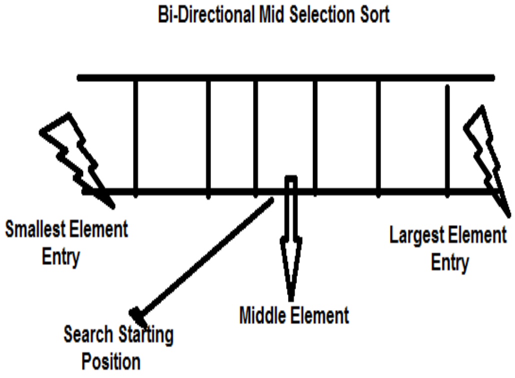
- Minimize the number of swapping
- Makes N-1 passes [ie Every pass, it excludes the maximum & minimum value location]
- High speed.

Limitations

- Occupies More memory space [ie need to declare two more variable for keeping former maximum and minimum value]

C. Bi-Directional Mid Selection Sort Algorithm

Sort the data by selecting the maximum & minimum elements from the middle to both sides in the selected list. This process decrements the size by 2 for each iteration. Two loops used for this process such as outer loop and inner loop. Outer loop manages the size of the list to be processed and inner loop is for identifying the maximum and minimum from selected list by outer loop [5]



Benefits

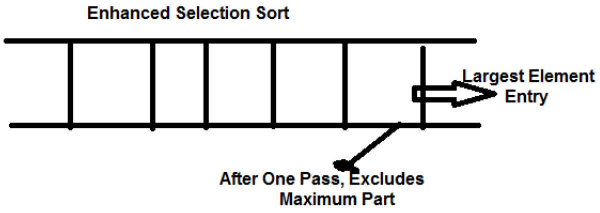
- Reducing the number of iterations
- Reduce Timing

Limitations

- Memory space for Next Maximum and Minimum
- Extra Time needed to find Middle element

D. Enhanced Selection Sort [ESS]

Introducing an array element and sorting the element by finding the maximum and swapping it with the largest element and decrease the size of the array by one for each pass. Before continuing with next maximum value, algorithm excludes the final component which was named in the previous measure. In each step, the list is to be reduced by excluding the last element for every step. It continues the process till the list [3]



Benefits

- N-1 comparisons
- High Speed
- If data is already sorted, ESS does not perform any swap operation

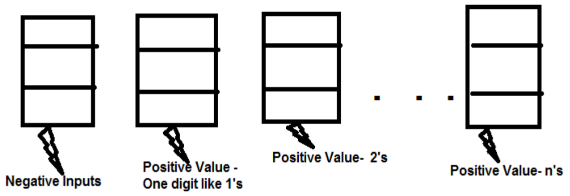
Limitations

- If the array is unsorted, SS & ESS performs the same

E. ARC Sort

Two dimensional data structure is used to store the input numbers belong to that bucket where one dimension represents the number of figures in each bucket and the second dimension represents the highest number of digits among the given input numbers ie.) each corresponding bucket's dimension represents the input numbers belong to that bucket. Numbers of digits are counted for each input bit. Store the number at the corresponding bucket. Increment the indexed array by number of digits. Repeat the process for each input number and occupy each number in its suitable position.

This process allot separate bucket for keeping negative numbers. Mostly first bucket is to use for negative values. Once all the number are placed in the corresponding buckets, now sort process proceeds individually using ESS [4]



Benefits

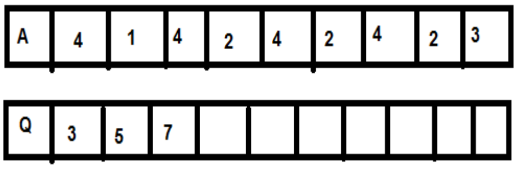
- Reduces the number of passing by dividing each element into buckets.

Limitations

- Memory Complexity is too high

F. Improved Selection sort Algorithm

This algorithm mainly focuses on repetition values in the list. It scans the duplications in a single pass and order the items as like a classical selection sort. The index can be used to maintain the repetitions location.[6]



Index maintains the 2's dublication location

Benefits

- Reduces run time
- Perform multiple scanning at single pass

Limitations

- Without repetitions, normal like Classical one
- Index Structure occupies more memory space.

IV. CONCLUSION

The selection sort algorithm is the simplest one to arrange the list of items in an effective manner. This paper compared the analysis of classical Selection Sort algorithm with Enhanced selection sort algorithms proposed by various researchers and reviewed the benefits and limitations in each algorithm. In future, these algorithms will be tested using a programming language with a variety of input values and analyses that performance based on timing and memory complexity.

REFERENCES

[1] Md. Khairullah "An Enhanced Selection Sort Algorithm", SUST Journal of Science and Technology, Vol. 21, No. 1, 2014

[2] Surender Lakra, Divya " Improving the performance of Selection Sort using a Modified Double Ended selection sorting", International Journal of Application or Innovation in Engineering & Management, Vol 2, issue 5, May 2013.

[3] Jehad Alnihoud, Rami Mansi, " An Enhancement of Major sorting Algorithms", The international Arab Journal of Information Technology, Vol 7, No. 1 January 2010.

[4] Ankit R. Chadha, Rishikesh Misal, Tanaya Mokashi, Aman Chadha " ARC Sort: Enhanced and Time Efficient Sorting Algorithm", International Journal of Applied Information Systems, Vol.7, No. 2, April 2014.

[5] Muhammad Farooq Umar, Ehsan Ullah Munir, Shafgat Ali Shad and Muhammad Wasif Nisar, " Enhancement of Selection, Bubble and Insertion sorting Algorithm", Research Journal f Applied Sciences, Engineering and Technology", 8(2) 263-271,2014.

[6] J.B.Hayfron-Acquah, Obed Appiah, K.Riverson, "Improved Selection Sort Algorithm" International Journal of Computer Applications, Vol. 110, No. 5, January 2015.