A Novel Sorting Algorithm and Comparison with Bubble sort and Insertion sort

NitinArora
Dept. of Computer Science & Engineering
G. B. Pant Engineering College, Pauri, Uttarakhand, INDIA

Vivek Kumar Tamta
Dept. of Computer Science & Engineering
G. B. Pant Engineering College, Pauri, Uttarakhand, INDIA

Suresh Kumar
Dept. of Computer Science & Engineering
G. B. Pant Engineering College, Pauri, Uttarakhand, INDIA

ABSTRACT
Sort is an algorithm that arranges all elements of an array, orderly. Sorting Technique is frequently used in a large variety of important applications to arrange the data in ascending or descending order. Several Sorting Algorithms of different time and space complexity are exist and used. This paper provides a novel sorting algorithm Counting Position sort which is based on counting the position of each element in array. We also compare Counting Position algorithm with Bubble sort and Selection sort. We have used the MATLAB for implementation and Analysis of CPU time taken for all the three sorting algorithms used. We have checked the algorithms with random input sequence of length 10, 100, 1000, 10000, 50000. Result shows that for the small length of input sequence the performance all the three techniques is all most same, but for the large input sequence Selection sort is faster than Bubble sort and Counting Position sort.

Keywords
Bubble Sort; Counting Position Sort; Selection Sort; Sorting.

1. INTRODUCTION
An algorithm is a finite sequence and well-defined set of computational instructions that takes some value or set of values as input and produces some value as a result [1]. A good algorithm is that which gives satisfactory result for every range of data set. Sorting is a very basic concept and important for solving other problems like is prerequisite for Binary Search. Sorting is the fundamental problem of computer science and remained burning issue for re-search over the last several years due to time complexity [2]. Sorting is often used in a large variety of critical applications and is a fundamental task that is used by most computers.There are two general categories of sorting algorithms: algorithms that sort random-access objects, such as arrays or random-access disk files, and algorithms that sort sequential objects, such as disk or tape files, or linked lists. There are three general methods for sorting arrays: [1]

- Exchange
- Selection
- Insertion

This paper provides a novel sorting algorithm Counting Position Sort and we compare Counting Position sort algorithm with Bubble Sort and Selection Sort.

The remainder of this paper is organized as follows. Section 2 introduces the overview of several existing sorting techniques; Section 3 describes our proposed counting position technique; Section 4 describes comparative study of our proposed technique with other existing techniques; Conclusion and future scope in Section 5; given acknowledgments in Section 6 and all the used references are given in section 7.

2. OVERVIEW OF SEVERAL SORTING TECHNIQUES

2.1 Bubble Sort
The bubble sort is an exchange sort. It involves the repeated comparison and, if necessary, the exchange of adjacent elements. The elements are like bubbles in a tank of water-each seeks its own level [1,3,4]. For example, if the Bubble Sort were used on the array, 9, 1, 10, 7, 3, 11, 2, 4, each pass would be like as shown in Table 1.

Table 1. Bubble Sort for the input values 9, 1, 10, 7, 3, 11, 2, 4

Initial	9	1	10	7	3	11	2	4
Pass1	1	9	7	3	10	2	4	11
Pass2	1	7	3	9	2	4	10	11
Pass3	1	3	7	2	4	9	10	11
Pass4	1	3	2	4	7	9	10	11
Pass5	1	2	3	4	7	9	10	11

With the Bubble Sort, the number of comparisons is always the same because the two for loops repeat the specified number of times whether the list is initially ordered or not. This means that the bubble sort always performs

$\frac{1}{2}(n^2 - n)$

Comparisons, where n is the number of elements to be sorted[4, 5].

2.2 Selection Sort
A selection sort selects the element with the lowest value and exchanges it with the first element. Then, from the remaining n-1 elements, the element with the smallest key is found and exchanged with the second element, and so forth. The exchanges continue to the last two elements [1, 3, 4]. For example, if the selection method were used on the array, 9, 1, 10, 7, 3, 11, 2, 4, each pass would be like as shown in Table 2.

Table2.Selection Sort for the input values 9, 1, 10, 7, 3, 11, 2, 4

Initial	9	1	10	7	3	11	2	4
Pass1	1	9	10	7	3	11	2	4
Pass2	1	2	10	9	7	11	3	4
Pass3	1	2	3	10	9	8	7	4
Pass4	1	2	3	4	10	9	8	7
Pass5	1	2	3	4	7	10	9	8
Pass6	1	2	3	4	7	8	10	9
Pass7	1	2	3	4	7	8	9	10

The selection sort requires

½ (n^2 – n)

Comparisons, where n is the number of elements to be sorted [4,5].

3. NOVEL PROPOSED COUNTING POSITION SORT ALGORITHM

Counting Position sort is a new sorting algorithm. It is based on counting the smaller elements in the array and fixes the position of the element. Counting Position sort uses the following algorithm:

```
Algorithm: Counting_Position_SORT(array, n-1) /* array is
set of total n input elements */
        for(i=1; i<=n;)
        {
                intcount=0; j=i+1;
                while(j<=n)
                {
                        if(array[i]>array[j]) then
                        count++;
                        j++;
                }/* end while*/
                        if(count>0) then
                swap(array[i] and array[ i + count]);
                        else
                        i++;
        } /*end for loop*/
```

For example, if the Counting Position method were used on the array, 9, 12, 10, 7, 3, 11, 2, 4, each pass would be like as shown in Table 3.

Table 3. Counting Position Sort for the input values9, 1, 10, 7, 3, 11, 2, 4

Initial	9	1	10	7	3	11	2	4
Pass1	11	1	10	7	3	9	2	4
Pass2	4	1	10	7	3	9	2	11
Pass3	7	1	10	4	3	9	2	11
Pass4	3	1	10	4	7	9	2	11
Pass5	10	1	3	4	7	9	2	11
Pass6	2	1	3	4	7	9	10	11
Pass7	1	2	3	4	7	9	10	11

4. COMPARATIVE STUDY AND DISCUSSION

All the three sorting algorithms (Bubble Sort, Selection Sort and Counting Position Sort) were implemented in MATLAB 8.0 and tested for the random sequence input of length 10, 100, 1000, 10000, 50000. All the three sorting algorithms were executed on machine with 32-bit Operating System having Intel(R) Pentium (R) CPU P6200 @ 2.13 GHz, 2.13 GHz and installed memory (RAM) 3.00 GB.

The Plot of length of input and CPU time taken (msec) is shown in figure 1. Result shows that for the small length of input sequence the performance all the three techniques is all most same, but for the large input sequence Selection sort is faster than Bubble sort and Counting Position sort.

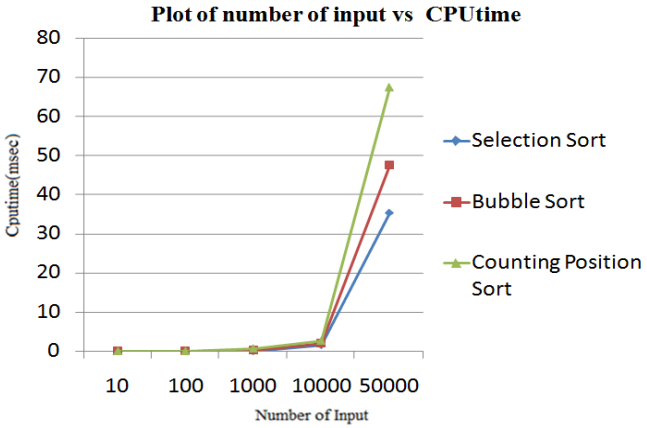

Fig.1 Plot of Number of Input vs CPU time(msec)

5. CONCLUSION AND FUTURE WORK

Sort is an algorithm that arranges all elements of an array, orderly. From the result it shows that new sorting algorithm is working well for all length of input values. But, it takes larger CPU time than Bubble sort and Selection sort.

In the future work we can make it more efficient in terms of execution time and can compare with other existing algorithm and can also calculate its time complexity.

6. ACKNOWLWDGEMENT

Our thanks to Dr. A.K. Swami, Principal, G. B. Pant Engineering College, Ghurdauri, for providing necessary infrastructure for the research work. We would also like to thank Mr. Sashi Kant Verma, Head, Department of Computer Science and Engineering, G. B. Pant Engineering College, Ghurdauri, for his unconditional and valuable support in writing this paper.

7. REFERENCE

[1] Herbert Schildt Tata McGraw-Hill [2005], "The Complete Reference C fourth Edition".

[2] Alfred V., Aho J., Horroroft, Jeffrey D.U. (2002) Data Structures and Algorithms.

[3] Frank M.C. (2004) Data Abstraction and Problem Solving with C++. US: Pearson Education, Inc.

[4] Cormen T.H., Leiserson C.E., Rivest R.L. and Stein C. (2003) Introduction to Algorithms MIT Press, Cambridge, MA, 2nd edition.

[5] Seymour Lipschutz (2009) Data Structure with C, schaum Series, Tata McGraw-Hill Education.