Sign Shuffle Sorting Algorithm (S_3 Algorithm)

Ajay Kedia
B-Tech IT Engineering
Vellore Institute of Technology
Vellore

Shagun Dhingra
B-Tech IT Engineering
Vellore Institute of Technology
Vellore

ABSTRACT

In computer Science, Sorting algorithm is used to place the elements in a certain order or to arrange the data of the array or string in a particular order. Sorted Data has been used to design the database which contains the array values in a certain order. Typically, Sorting Algorithm is used to arrange the data either in increasing order to decreasing order. There are certain types of sorting algorithm which are used to sort an array and have different cost and time complexity but we have discussed only new sorting algorithms designed by us. Comparison of our sorting algorithm with others sorting algorithm are done in this paper. Because sorting is important to optimize the use of other algorithms in computer science such as binary search or linear search, it has been the subject of extensive research in computer science, and some complexity methods have been developed.

General Terms

Algorithms

Keywords

Sorting, Shuffling, Sign, Swapping, Exchanging, Divide, Processors.

1. INTRODUCTION

We are going to introduce another type of algorithm which sorts an array on the basis of their sign and shuffle the elements.

1.1 Introduction to Sign Shuffle Sorting Algorithm

As we are going to introduce a new algorithm, there are some features on which we give our attention the most. First, we are checking the first element with the last element and then second element with the second last element and so on. Then second, we are applying divide and conquer rule. We divide the array into two equal parts and perform the first step individually on the divided parts. This process carries on till further division is not possible .Then; we use the concept of sign.

2. RELATED WORK

2.1 Data Representation

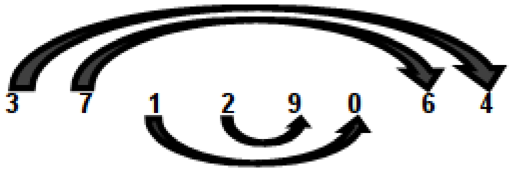
In Sign Shuffle Sorting or S_3 Algorithm, data can be a string, float, integer, double or long. An array contains all these data. Using the loop, we can store the data into an array. It is a simple algorithm by which we can sort millions of data on the basis of their sign.

A user who wants to get the details of his/her transaction on the basis of increasing order of the date of his/her withdrawal, a web designer wants to store the data in increasing or decreasing order in his/her database, so everywhere we are using the concept of sorting.

2.2 How does Sign Shuffle Algorithm work?

Suppose we are having some data in an array: - 3 7 1 2 9 0 6 4

A user wants to sort the data. Therefore by using the Sign Shuffle Sorting Algorithm, we are going to sort this array.



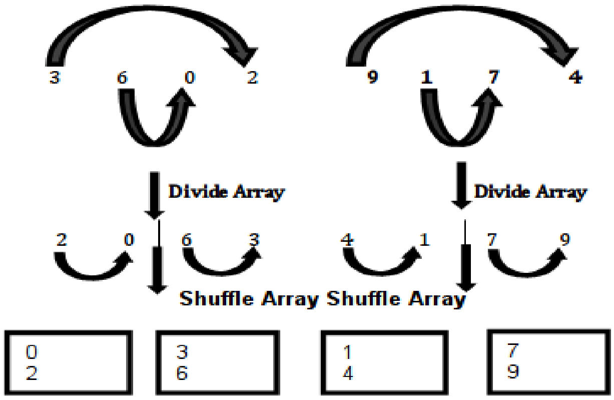
Compare the first element with the last element and so on. If the first element is larger than the last one, shuffle it otherwise no shuffle. Do the same thing with the other elements. After this, our array will be:

3 6 0 2 9 1 7 4

Now, divide the array into two equal parts, here as length of the array is 8 so divide it into 2 equal parts, if the length of the array is odd then divide as much that the divided array's length is differed by 1.

3 6 0 2 | 9 1 7 4

Now we are having two arrays, one is **3 6 0 2** and another one is 9 **1 7 4** and do the same thing as we did above. Compare the first element with the last and so on.

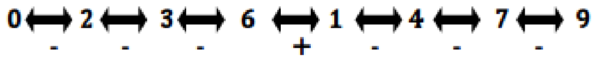


After combining these arrays, the final array will be:

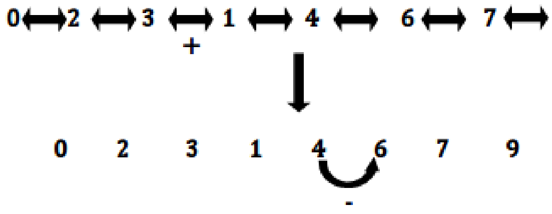
0 2 3 6 1 4 7 9

2.3 What’s the sign means in Sign Shuffling Algorithm?

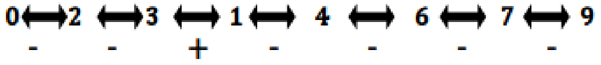
In our algorithm, we are using a term sign which will be used to compare the value of one element with its adjacent element. If the value of the element is greater than the adjacent element, we assign a positive sign otherwise give negative sign and do the same with other elements.



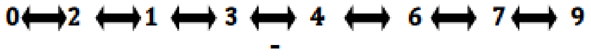
All Signs should be negative if we want to sort the array into an ascending order otherwise should be positive for the descending order. Here, we are going to sort the array in an ascending order. Shuffle the elements which have positive sign and after shuffling, make it a pivot and change the sign of furthermost element and reshuffle the shuffled element until it gives a –ve sign.



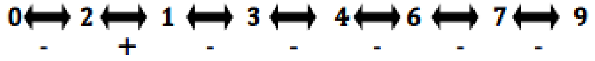
As 6 gets his right position, now again find out the sign of the other elements.



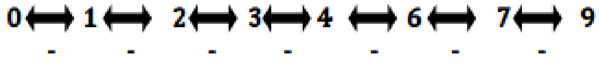
Now shuffle the value of **3** and **1** and adjacent element will compare its sign with the shuffled element and again the positive sign then again shuffle and so on.



Now the sign is negative so we will not shuffle it further. Now carry out the same steps again.



Again shuffle the value and their sign after shuffling



Now all signs are negative. Hence, our array is sorted in ascending order. We can sort a large number of data.

In parallel computing we can use n/2 processor to carry out the task. And in sign shuffle we can use only one processor because the furthermost elements depend upon the previous exchange.

0 1 2 3 4 6 7 9

This whole concept can be understand by below flow chart

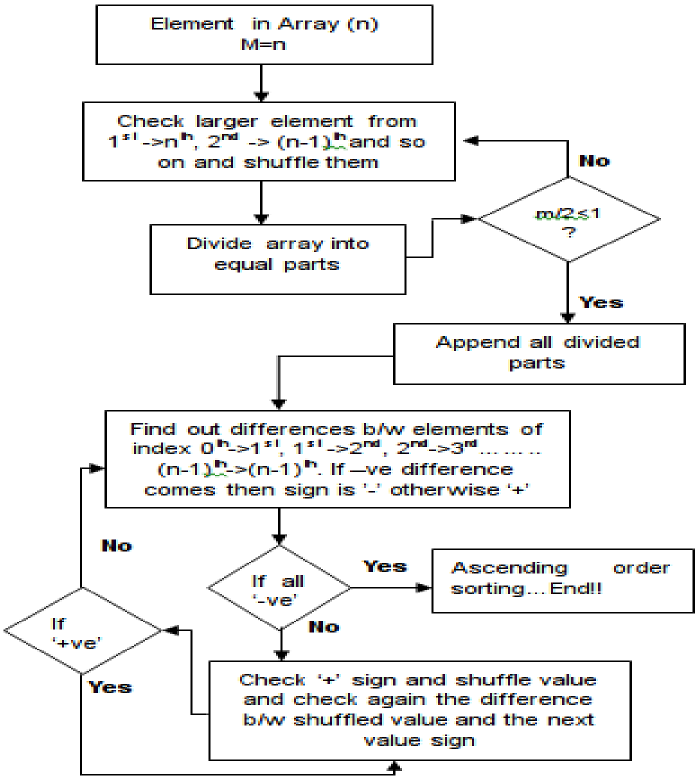

Element in Array (n)
M=n
Check larger element from 1st->nth, 2nd -> (n-1)th and so on and shuffle them
Divide array into equal parts
m/2≤1 ?
No
Yes
Append all divided parts
Find out differences b/w elements of index 0th->1st, 1st->2nd, 2nd->3rd……..(n-1)th->(n-1)th. If –ve difference comes then sign is '-' otherwise '+'
If all '-ve'
Yes
Ascending order sorting…End!!
No
Check '+' sign and shuffle value and check again the difference b/w shuffled value and the next value sign
If '+ve'
Yes
No

Fig 1. Flow chart for ascending order sorting

3. RESULT

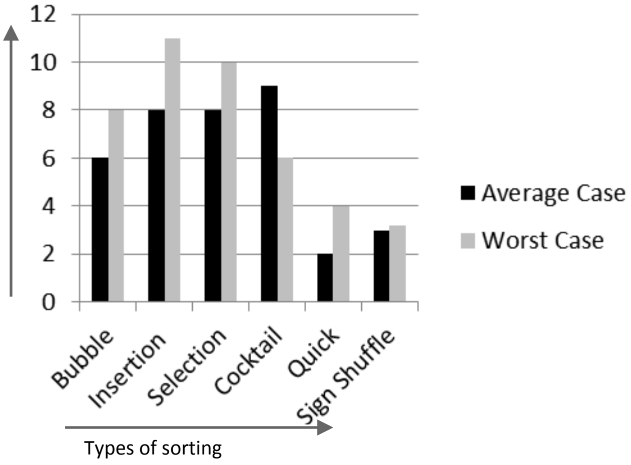
S3 Algorithm has been checked with certain number of sorting

Table 1. Comparison of number of swaps which relates to several sorting algorithm

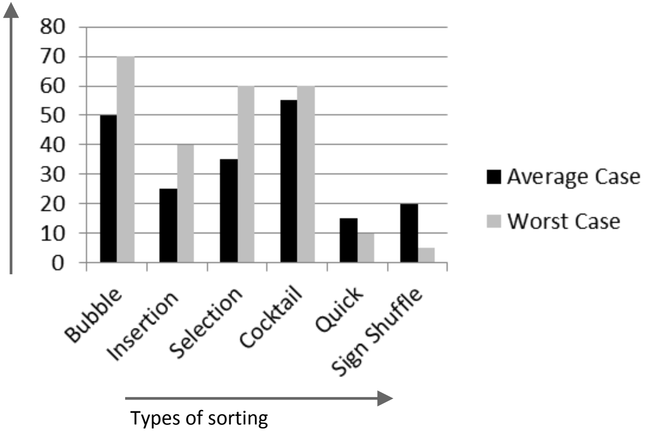
Sort	N=100	N=1000	N=10000
Bubble Sort	7890	879604	98530146
Insertion Sort	3146	243578	23453685
Selection Sort	4913	267548	35648754
Cocktail Sort	4987	574970	45367578
Sign Shuffle Sort	3019	227492	21348636

**Note: N= the number of integers in an array.

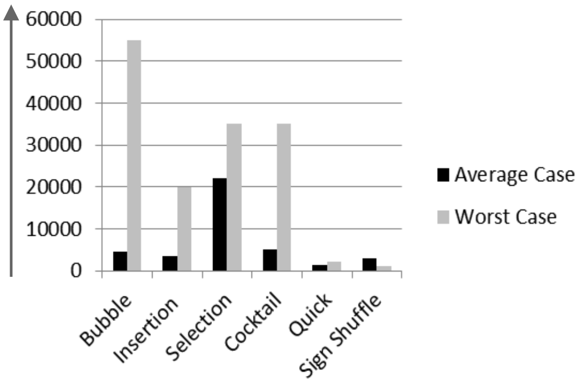
The relationship between average and worst cases for sorting for N = 100



The Relationship between runtime in average and worst case for several Sorting for N = 1000



The Relationship between runtime in average and worst case for several Sorting for N = 10000



4. PSEUDO CODE

4.1 Pseudo code of Sign Shuffle Sorting Algorithm

```
Step 1:
int a[n];
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);     // storing the elements into an array
}
j=n-1;
m=n;
Step 2:
for(i=0;j>n/2 && i<n/2;i++,j--)
{
if(a[i]>a[j])    //comparing the first value with the last one and so on
{
swap(a[i],a[j]);   // if value is greater than we will swap the value
}
}
Step 3:
Divide the array into half and repeat step 2 for both the divided parts;
Step 4: repeat the step3 until m/2=1;
Step 5:
int temp;
for(i=0;i<n;i++)
{
if(i+1<n)// for the last value , otherwise it  will show an error
temp=a[i]-a[i+1];  // difference between two adjacent values
if(sign(temp)>0) //positive and negative sign checking, as ascending order case
swap(a[i],a[i+1]) // swap the elements if +sign is there continuously until or unless negative sign does not come.
}
Step 6:
 repeat step5 until all sign of the difference is negative.
Step 7: end
```

5. CONCLUSION

Sign shuffle Sorting is used to sort the elements on the basis of the sign of the difference of the elements with their adjacent elements. If we calculate the time complexity of the sorting, in Step 2 it will take O($\log_2$n) time and in step 3 to step 6 it will take only O($\log_2$n) time. So the total time complexity is O($\log_2$n).

Based on our experiment, we have calculated the average and worst time complexities of several sorting algorithms and compared it with sign shuffle sorting algorithm and we have found that sign shuffle sorting is just close to quick sorting but in average case, quick sorting algorithm shows the better graph than sign shuffle sorting. Worst case of sign shuffle sorting will be performing sign shuffle sorting on the elements which are already arranged in sorted order. In that case our flow chart will first check all the signs, then after it will move to shuffling if there is any positive sign. So, in worst case our sorting takes minimum runtime. And hence, we got a smaller graph in this case among all the sorting.

6. ACKNOWLEDGMENTS

We express our sincere thanks to www.Google.com help me through out and provide me the information which I was needed .A heart full and sincere gratitude to our parents who motivate us through out and help us in our bad time.

7. REFERENCES

[1] Lavore, Robert, arrays, Big O Notation and simple sorting. Data structure and algorithms in Java (2nd edition)

[2] Simple algorithm example. http://www.cs.cmu.edu/_adamchik/15-121/lectures/Sorting%20Algorithms/sorting.html

[3] Wikipedia, the free encyclopedia. Online: Sorting algorithms. http://en.wikipedia.org/wiki/Sortingalgorithms