ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 3, January 2018

CS SORTING ALGORITHM

S.Manjunath, Chandrashekar.C,
Computer Science and engineering,
S.J.C Institute of Technology
Chikballapura-572101

ABSTRACT

Algorithms are step by step procedure to solve a given problem. Out of many problems out there sorting has attracted manyresearchers mainly due to its complexity. In this paper we will be discussing one such algorithm. The name of the algorithm is CSsorting. The proposed algorithm uses the decrease and conquer technique to sort an array on A[0…n-1]..

***KEYWORDS**: Algorithm, Double-linked list, Time complexity*

I. INTRODUCTION

The sorting algorithm in this proposed paper represents a new method in sorting. Sorting is a technique where the given data isarranged logically based on the user's requirement (ascending/decending). Efficiency ofa any algorithm depends upon two main factors :

1) Time complexity
2) Space complexity

There are many techniques available to sort a given problem here we are going to use decrease and conquer. The decrease andconquer technique is based on exploiting the relationship between a solution to a given instance of a problem [1].

Even though there are many sorting algorithms that are effective. Even though many researchers are working to find out newmethods to solve this particular algorithm they are largely ineffective. Here we have tried to design a algorithm that is far more effectivethan many algorithms that are present today

II. DATA STRUCTURES USED IN ALGORITHM

The data structure used in our algorithm is doubly linked lists. [2]This particular data structure is used when we have to move ineither direction or in which we must delete a arbitrary node. Each node has two link fields, one linking in the forward direction and theother in the back-ward direction. A node in a doubly linked list has three fields, a left link field, a data link field and a right link field. Thenecessary declarations are:

```
typedef struct node * nodePointer ;
typedef struct {
nodePointer llink ;
element data ;
nodePointer rlink ;
} node ;
```

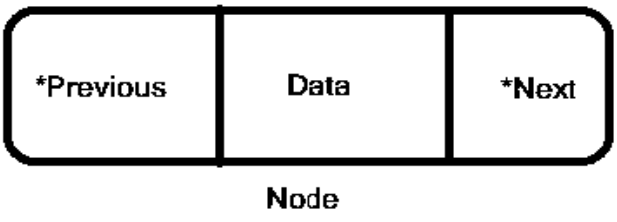


Figure1 shows a node of doubly linked list

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 3, January 2018

III. PREVIOUS WORKS

Some well known sorting algorithm are bubble sort, selection sort, insertion sort ,merge sort, quick sort ,heap sort ,radix sort ,cocktail sort ,shell sort etc. All these algorithms can be classified according to their average case and worst case time complexity(asymptotic complexity-big theta notation-θ and big oh notation – O respectively). Time complexity of some algorithms are given in Table1 along with the stability of these algorithms i.e. they are stable or not. Here (with respect to the Table 1) n is the input size of the list to besorted, d is number of digit in the largest number among inputs and k is all possible digit/word (for example- k=10 as decimal). The timecomplexity of some well known algorithm are mentioned in figure 2.

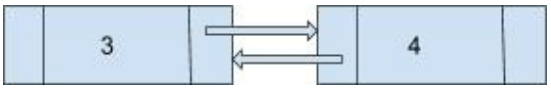
Name of the algorithm	Time complexity:Best	Time complexity:Worst	Stable ?
Bubble sort	Θ(n^2)	Θ(n^2)	Yes
Selection sort	Θ(n^2)	Θ(n^2)	Yes
Insertion sort	Θ(n^2)	Θ(n^2)	Yes
Merge sort	Θ(n log2 n)	Θ(n log2 n)	Yes
Quick sort	Θ(n log2 n)	O(n2)	No
Heap sort	Θ(n log2 n)	Θ(n log2 n)	No
Bucket sort	Θ(d(n+k))	O(n2)	Yes

Figure 2 shows the best and worst cases of existing algorithm

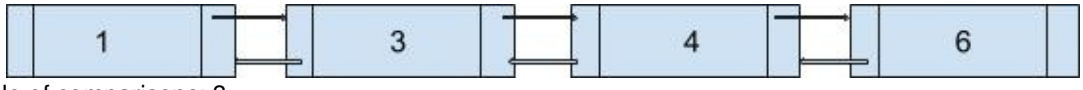
IV. PROPOSE ALGORITHM

The algorithm for the proposed sorting techniques is given below:
Step 1: Input the n number of elements into the array.
Step 2: Choose the 0 th and 1 st element from the array and compare both the elements and place it on the doubly linked list as per user'srequirement (increasing/decreasing).
Step 3: Now choose the i th and (i+1) th elements where, i should be less than n and also greater than 1.
Step 4: Apply binary search on the doubly linked list to find suitable positions for the elements selected.
Step 5: Once the suitable positions place the elements in their respective positions by creating new nodes.
Step 5: Repeat step 3, step 4 and step 5 until all elements are sorted in the doubly linked list.
Step 6: End.

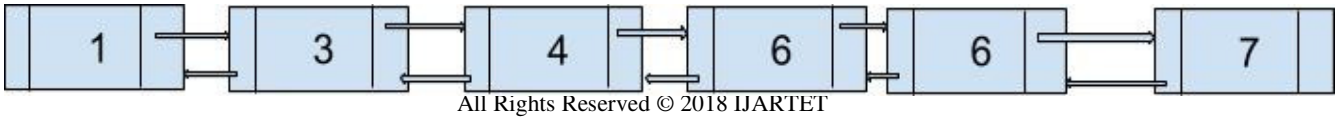
Example for the proposed algorithm
I/p array: A[5] ={4,3,6,1,6,7};
First pass:



No of comparisons: 1
Second pass:



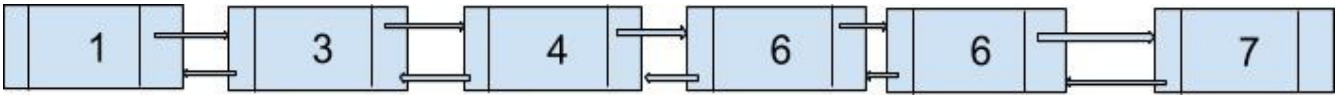
No of comparisons: 3
Third pass:



ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 3, January 2018

No of comparisons: 7
Sorted List



V. APPLICATIONS

A solution to a given problem is of no use if we don't find a suitable application that uses the solutions effectively. The algorithm proposedcanbeusedinmanyreallifesituations,someofthemare mentionedbelow:

A.Sortinginthemusicinamusicplayer

When new music are added to a music player application the new content can be added into the list as music players rely on doublelinkedlisttomovebetweensongs.

B.Sortinginatailoringshop

This algorithm can be used in a tailor shop, how you ask? when new cloths are brought in, the tailor can use the proposed sorting techniquetosortthenewlycomeclotheswiththeoldones.

VI. CONCLUSION

The presented algorithm is still in yet to be further studied so that we can implement it successfully. The advantage of this particularalgorithmisthetimerequiredtosort,stability.Selectinga sortingalgorithmispurelybasedonthegivenproblem.

There is a big scope for this proposed algorithm is the future and we have to experiment a lot with the proposed algorithm to find outsitsshortcomings(if-any)bygivingitunconventionalreallifeinputs.

VII. REFERENCES

[1]AnanyLevitin,"Introductiontothedesignandanalysisofalgorithms2nd edition,ISBN:9780321358288
[2]HorowitzandSahaniandAndersonFreed,"FundamentalsofdatastructuresinC"2nd edition,ISBN:9788173716058