Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

An Optimized Input Sorting Algorithm

Anshu Mishra[1] and Garima Goyal[2]

[1] Jyothy Institute of Technology/ Visvervaraya Technological University

Received: 13 December 2015 Accepted: 3 January 2016 Published: 15 January 2016

Abstract

One of the fundamental issues in compute science is ordering a list of items. Although there is a huge number of sorting algorithms, sorting problem has attracted a great deal of research, because efficient sorting is important to optimize the use of other algorithms. Sorting involves rearranging information into either ascending or descending order. This paper presents a new sorting algorithm called Input Sort. This new algorithm is analyzed, implemented, tested and compared and results were promising.

Index terms— algorithm, sorting, input sort, insert.

1 Introduction

nformation growth rapidly in our world and to search for this information, it should be ordered in some sensible order. Many years ago, it was estimated that more than half the time on many commercial computers was spent in sorting. Fortunately this is no longer true organizing data, methods which do not require that the data be kept in any special order [1].

Many algorithms are very well known for sorting the unordered lists. Most important of them are Bubble, Heap, Merge, Selection [2]. As stated in [3], sorting has been considered as a fundamental problem in the study of algorithms, that due to many reasons: 1. The need to sort information is inherent in many applications. 2. Algorithms often use sorting as a key subroutine. 3. In algorithms design there are many essential techniques represented in the body of sorting algorithms. 4. Many engineering issues come to the fore when implementing sorting algorithms. Efficient sorting algorithms is important to optimize the use of other algorithms that require sorted lists to work correctly; it is also often in producing human readable output. Formally, the output should satisfy two major conditions: 1. The output is in non-decreasing order. 2. The output is a permutation or reordering of the input. Since the early beginning of computing, the sorting problem has attracted many researchers, perhaps due to the complexity of solving it efficiently. Bubble sort was analyzed as early as 1956 [6].

Author ? ? : Assistant Professor, Department of Information Science & Engineering Jyothy Institute of Technology, Bangalore. e-mails: anshu.garg13@gmail.com, goyal.garima18@gmail.com introductory computer science classes, where the abundance of algorithm for the problem provides a gentle introduction of core algorithm concepts [4,5]. In [4], they classified sorting algorithms by: 1. Computational complexity **??**

2 Input Sort a) Concept

A simple sorting algorithm which sort the data whenever it is input from any input source e.g. keyboard or data from a stream of file. when new item comes then it is inserted at its specific position through a recursive function if there are n elements then n items 1 at a time is inserted in array which increase array size automatically and take its appropriate position.

3 GET_INPUT( ) for

i = 1 to n { scan(c[i]) call INPUT_SORT(c[i],1,size+1) } end III.

4 Working

5 Suppose we have array c[1?5] of five elements as follows:

First we call GET_INPUT() function and read input from input source Which is Sorted Array .

IV.

6 Complexity

Generally the complexity of an algorithm is measured in two phases. When one measures the complexity of an algorithm by pen and paper, he/she can only predict the complexity which give an idea how much time and space this algorithm takes to finish in its execution. This phase is called priory analysis. After implementing the algorithm in computer, we get the actual time and space. This phase of analyzing the algorithm is called the posterior analysis. complexity of an algorithm can be of two types: 1. Time Complexity: The analysis of algorithm for the prediction of computation time for execution of each and every instruction in the algorithm is called the time complexity. [1]

Figure 1:

[1] © 2016 Global Journals Inc. (US)

Some
sorting algorithms are ”in-place”, such that only
O(1)or O(log n)memory is needed beyond the items
being stored, while others need to create auxiliary
locations for data to temporally stored.
3. Recursion some algorithms are either recursive or
non recursive, while others may be both (e.g merge
sort).
4. Whether or not they are a comparison sort. A
comparison sort examines the data only by
comparing two elements with a comparison
operator.
This paper presents a new sorting algorithm
called input sort. Its typical use is when sorting the
elements of a stream from file.
II.

Figure 2:

An Optimized Input Sorting Algorithm
Year 2016
12
Volume XVI Issue I Version I
( )
Global Journal of Computer Science and Technology
Using the standardrecurrenceequ
T(n)=aT[n/b]+f(n) get this equation:
T(n)=2T[n/2]+n a=2 b=2 f(n)=n
n log a log 2 b = n 2 = n
using master method's 2 nd case apply

[Note: HSortingBest Case]

Figure 3:

if f(n)=? (n log a b), then T(n)= ? (n log a b . log n) Time complexity of Input Sort is T(n)= ? (n logn)

.1 VI. comparison with heap and merge sort

Now if we talk about heap merge sort than our algorithm is better from two in the sense that In merge sort we need two extra temporary array which increase its space complexity but no need of extra memory in our algorithm. our algorithm has order of O(nlog n) but it execute fast because of less comparisons than Merge heap and Quick sort.

.2 VII.

.3 Conclusion

In this paper new sorting algorithm is presented INPUT-SORT has O(nlog n) complexity but it is faster than existing sort mentioned in section 4 in detail. INPUT-SORT is definitely faster than other sort to sort n elements. Furthermore, the proposed algorithms are compared with some recent sorting algorithms; selection sort and bubble sort, heap, merge, insertion, quick sort. These algorithm can be applied on a real world application. any sorting algorithm might be a subroutine of another algorithms which affects its complexity.

[Astrachanm ()] *Bubble Sort: An Archaeological Algorithmic Analysis*, O Astrachanm . 2003. Duk University

[Shahzad and Afzal ()] 'Enhanched Shell Sorting Algorithm'. B Shahzad , M Afzal . *computer general of Enformatika* 2007. 21 (6) p. .

[Cormen et al. ()] *Introduction to Algorithms*, T Cormen , C Leisersion , R Rivest , C Stein . 2001. McGraw Hill.

[Thoroup ()] 'Randomized Sorting in O(n log log n) Time And Linear Space Addition, Shift, and Bit Wise Boolean Operations'. M Thoroup . *Computer Journal of Algo rithms* 2002. 42 (2) p. .

[Knuth ()] *The Art Of Computer Programming*, D Knuth . 1998. Addison Wesley.

[Aho and Hopcroft ()] *the design and analysis of computer Algorithms*, A Aho , J Hopcroft , UllmanJ . 1974. Addison Wesley.