www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 5 Issue 09 September 2016 Page No.17971-17972

MM Sort algorithms

Mihir Narshana
mmnarshana1996@gmail.com
BE student Vadodara Institute of Engineering
Vadodara

Abstract

Sorting is used for arranging a data either in ascending order or descending order. I have discussed here in these paper about new Algorithm to solve the problem of sorting. The Algorithm to solve problem of sorting is on reference of BUBBLE sort technique. The procedure to sort the element is extremely different.

Introduction

Algorithm is an unambiguous, step-by-step procedure for solving a problem, which is guaranteed to terminate after a finite number of steps.[5][4] In other words algorithm is logical representation of the instructions which should be executed to perform meaningful task. For a given problem, there are generally many different algorithms for solving it. Some algorithms are more efficient than others, in that less time or memory is required to execute them. In simple word it is the method to arrange the desired elements into a sequential order.[4] We have come across many things where we need sorting in our day to day life. E.g.-

1) In telephone diary
2) Roll no in merit list
3) Any particular page in a book, and many more.

I. Bubble sort:-

The bubble sort is the oldest and simplest sorting method in use. Unfortunately, it's also the slowest. The bubble sort works by comparing each item in the list with the item next to it, and swapping them if required. The algorithm repeats this process until it makes a pass all the way through the list without swapping any items (in other words, all items are in the correct order). This causes larger values to "bubble" to the end of the list while smaller values "sink" towards the beginning of the list. The total number of comparisons, is (n - 1) + (n - 2)...(2) + (1) = n(n - 1)/2 or O(n 2). The bubble sort is generally considered to be the most inefficient sorting algorithm in common usage. Under best-case conditions (the list is already sorted), the bubble sort can approach a constant O(n) level of complexity. General-case is an abysmal O(n 2). While the insertion, selection, and shell sorts also have O(n 2) complexities, they are significantly more efficient than the bubble sort. [7]. Don Knuth, in his famous The Art of Computer Programming, concluded that "the bubble sort seems to have nothing to recommend it, except a catchy name and the fact that it leads to some interesting theoretical problems", some of which he discusses therein. Bubble sort is asymptotically equivalent in running time to insertion sort in the worst case, but the two algorithms differ greatly in the number of swaps necessary.

II. Working Procedure Of Algorithms

The Bubble Sort is the simplest sorting technique, in which smallest data element is moved or „bubbled up" to the top. In this method, first element is compared with the next element in the array. So with reference to bubble sort I am applying my own technique.

DOI: 10.18535/ijecs/v5i9.30

Instead of direct swapping of two elements like as in bubble sort we are going to concatenate first two element and then dividing the concatenated by 100.

Now after dividing we would compare the quotient and the reminder. If the value of reminder I greater then quotient then the quotient and reminder gets swapped. Else the process continues for the next two elements.

III. **Proposed Algorithm**:-

MM sort {a[ ], I, j, quotient, reminder}

1) Enter the number of elements you want to sort.
2) Insert the elements.
3) Firstly concatenate first two elements entered by the user.

   //i.e. a[i]+a[i+1] gets concatenated.

4) Divide the concatenated integer value by 100.
5) If (quotient<reminder)
   Swap (quotient, reminder)
   Else
   Proceed for the next two elements.
6) Follow the procedure until whole array elements gets sorted.

IV. **Future scope**

In case if we become successful for finding alternate to perform division method by the present known method then these would definitely result into best algorithm for sorting.

Reference

1. A Comparative Analysis of Sorting Algorithms on Integer and Character Arrays .Ahmed M. Aliyu, Dr. P. B. Zirra . ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805
2. A Comparative Study on Different Types of Sorting Algorithms (On the Basis of C and Java). Rekha dwivedi PG scholar Department of Computer Science & Engineering SVITS –Indore (M.P) Email: rekhadwivedi40@yahoo.com
3. Owen Astrachan, Bubble Sort: An Archaeological Algorithmic Analysis, SIGCSE,2003, http://www.cs.duke.edu/~old/papers/bubble.pdf.
4. Comparative Analysis & Performance of Different Sorting Algorithm in Data Structure Miss. Pooja K. Chhatwani, Miss. Jayashree S. Somani Lecturer, Information Technology, Dr. N. P. Hirani Institute of Polytechnic, Pusad, Maharashtra, India
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest , Introduction to Algorithms, Fifth Indian printing (Prentice Hall of India private limited), New Delhi-110001
6.]Source:- http://www.codecodex.com/wiki/index.php?title=Bubble_sort# Performance .
7. Boolos, George & Jeffrey, Richard (1974, 1980, 1989, 1999), Computability and Logic (4th ed.), Cambridge University Press, London, ISBN 0-521-20402-X: cf. Chapter 3 Turing machines where they discuss "certain enumerable sets not effectively (mechanically) enumerable"