ANALYSIS OF BUBBLE, INSERTION, SELECTION AND MERGE SORT

Akriti Gupta[1], Parul Choudhary[2]

B.E. Student Department of CSE, KITE, Raipur

Asst. Prof. , Department of CSE, Kruti Institute of Technology & Engineering, Raipur [C.G], India

Email Id: cparul2605@gmail.com

ABSTRACT

Arranging elements in a specific order is called sorting. There are mainly two orders are generally in used like lexicographical and numerical order. For optimizing use of other algorithms efficient sorting is important such as insertion, selection, merge and bubble sorting algorithms. In this paper we analyze this different type of sorting with the help of algorithms and examples. One always requires data input which have to be sorted in specific order and produces output in human readable form. But it has to satisfy two conditions – first it should be in non-decreasing order and second it is in re-ordering order but with original elements of all inputs.

Keywords: - Bubble, Insertion, Selection, Merge sorting, Pseudo codes.

1. Introduction

Sorting is the technique to arrange the array elements into ascending or descending order. We can also stores the elements into the array. By using sorting we can sort the numbers, alphabets and records. Sorting can be done by any method which is suitable for it. There are many different types of sorting techniques in Computer Science to sort or stores the values into the array, they techniques are Bubble sort, Insertion sort, Selection sort and merge sort . Sorting arranges data in a sequence which makes searching easier. The sorting by shifting one by one element into the array is known as Insertion sort. By selecting a key and compares it the elements which is ahead of this element is known as Selection sort. The list given for sorting can divide into various sub strings and after sorting we merge it this type of sorting is known as Merge sort. Bubble sort is the technique of sorting in which we compares the elements into pairs and then swap it. By using sorting we can find our data stored into the array is easier way.

A. Application:

1 Now a day's people usually prefer sorted data to read.

2 A sorted array is much easier to search the data stored in array.

3 Sorting makes it much easier to discover patterns or statistics of data items, such as the median or other moments.

4 Sorting often helps in comparing lists (or sets), and performing operations like intersection of sets, finding out if two lists contain the same elements, finding if there are duplicates in a list, etc.

B. Limitations:

1 The maximum length of a sort key in a GROUP BY clause, ORDER BY clause, SELECT DISTINCT statement, or outer join is 255 bytes; the maximum length of all sort keys in a sort row is 65,500 bytes.

2 The loop continues to run even if the array is sorted if the code is not optimized.

2.Description

Let us discuss all four sorting i.e. bubble, insertion, selection and merge sorting one by one.

A. Bubble Sort: The idea of repeatedly comparing pairs of adjacent elements and if they arrange in wrong order then swap their place is known for bubble sort.

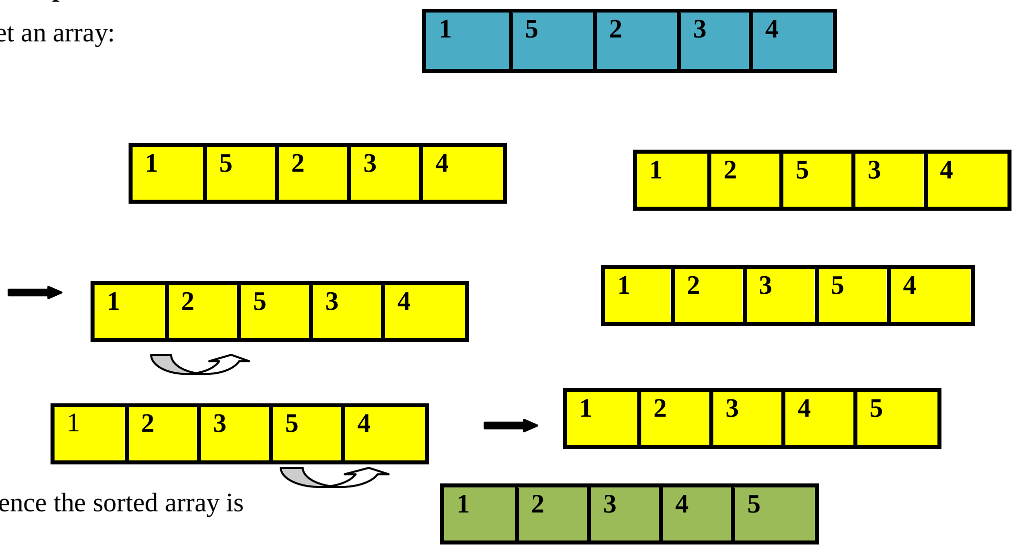
Suppose we have B[ ] is unsorted array of m number of elements. Now we want all ements to be sorted in ascending order.

Implementation --

```
void Sort_Bubble ( int B[ ], int m)
{
int asc;
for  (int i=0;i<m-1;i++)
 {
for(j=0;j<m-i-1;j++)
 {
if (B[j] > B[j+1])
{
   asc= B[j];
   B[j]= B[j+1];
   B[j+1] = asc;
} } } }
```

Example --

Let an array:



Hence the sorted array is

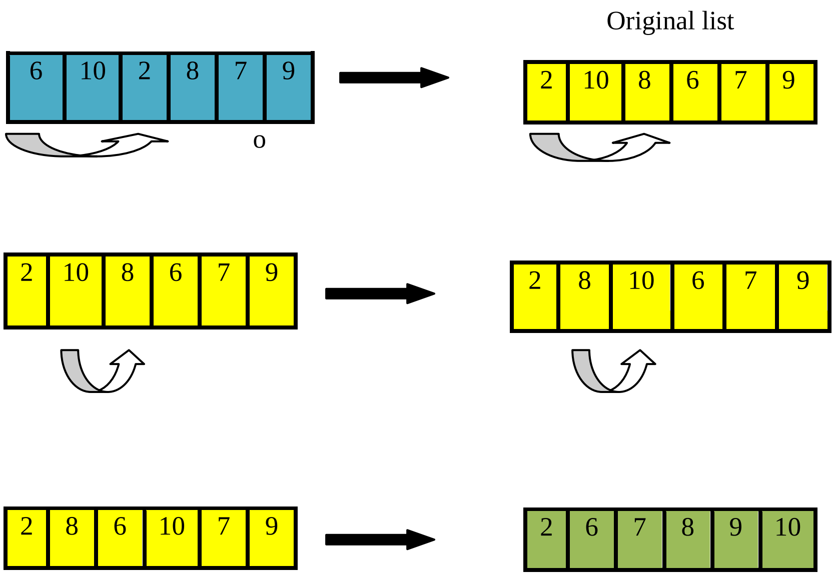
Fig 1. Example for Bubble Sort

B. Selection Sort: In an unsorted array, the idea of finding the minimum or maximum element then arrange its in proper position is known for Selection sort. Implementation is like.

Implementation

```
void Sort_Selection (int B[ ], int m)
{
        int min;
        for (int k = 0; k < m-1 ;  k++)
{
            max = k ;
            for (int j = k+1; i < m; i++ ) {
                if (A[ i ] < A[ min])  {
                min = i ;
                }  }
           swap ( A[ min ], A[ k]) ;
        }    }
```

Example --



Hence the sorted array found

Figure 2: Example for Selection Sort

C. Merge Sort: It is based on the idea of breaking down a list into several sub-lists until each sub list consists of a single element then merge these into a sorted list. It is also known for divide-and-conquer algorithms.

Implementation ---

```
void Sort_Merge ( int B [ ] , int BEG, int MID, int END){
 int a = BEG ,b = MID+1;
int Array [END-BEG+1] , k=0;
for ( int i = BEG; i <= END; i++)
 {
    if (a > MID)
       Array [ k++ ] = B [ b++] ;
 else if ( b > END)
       Array [ k++ ] = B[ a++ ];
else if( B[ p ] < B[ b ])
      Array [ k++ ] = B[ a++ ];
 else
      Array [ k++ ] = B[ b++];
 }
  for (int a=0 ; a< k ;a ++) {
     B [ BEG++ ] = Arr[ a ] ;
 } }
```

Example ---

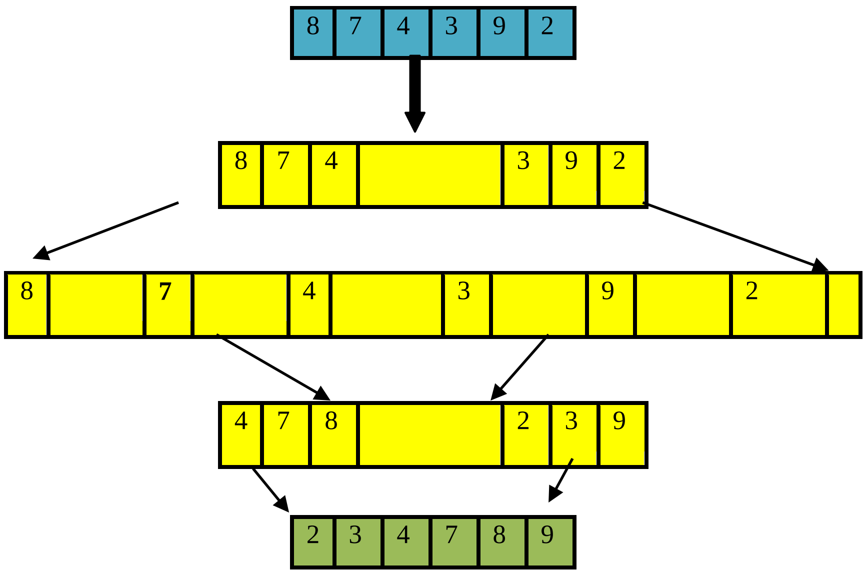


Figure 3: Example for Merge Sort

D. Insertion Sort: The idea is based on for finding element's correct position; one element from the input elements is consumed in each iteration to find its correct position is known for Insertion sort.

Implementation –

```
void Insertion_Sort ( int D [ ] , int m) {
     for ( int k = 0 ;k < m ; k ++ )
{
          int temp = D [ i ];
      int i = k;
          while (  i > 0  && temp < D [ i -1]) {
                D [ i ] = D [ i-1];
                i= i - 1;
           }
           D [ i ] = temp;
     }  }
```

Example ---

Let consider an array:

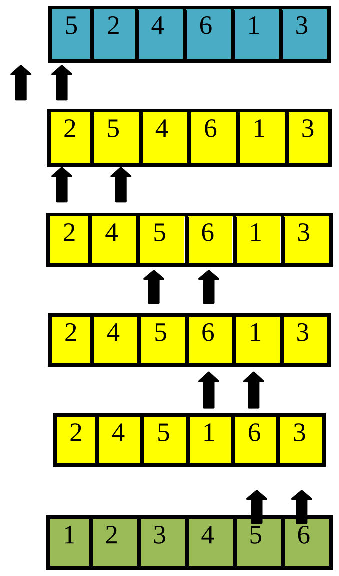


Figure 4: Example for Insertion Sort

3.Comparative Analysis

We can analyze all sorting and elaborate it with the help of table given below.

Complexity: In terms of computer science complexity is used to measure the required recourses for the execution of an algorithm. Complexity of an algorithm is a calculation of the amount according to their time and/or space required by an algorithm for an input of a given size (n).

Space Complexity: Required memory space for working of an algorithm is the Space complexity. A sorting algorithm has O (1) by allocating the working storage , such as a few variables for iteration and that is not proportional to the size of the input.

Time Complexity: Time complexity of an algorithm defines the total time required by the algorithm to perform its task till its completion. The time complexity of algorithms expressed by using big O notation. The time complexity is estimated by counting the number of elementary functions performed by the algorithm.

Worst case complexity: Analyzing the efficiency of an algorithm in the worst case tell us about how fast the maximum run time grow when we increase the input size of an array.

Best case complexity: In best case input supplied to the algorithm is much similar to the output format which is expected. In best case we can computed by performing dry run on an algorithm.

Average case complexity: Efficiency of an algorithm tell us about probabilistic analysis by which we find expected runtime for an algorithm. The time complexity that we get certain set of inputs is as a average same.

Complexity/Sorting	**Insertion Sort**	**Merge Sort**	**Selection Sort**	**Bubble Sort**
Worst –case	O(n^2)	O(n log n)	O(n^2)	O(n^2)
Best case	O(n)	O(n log n)	O(n^2)	O(n)
Average case	O(n^2)	O(n log n)	O(n^2)	O(n^2)
Space Complexity	O(1)	O(n)	O(1)	O(1)

Table 1: Comparative analysis of all sorting in terms of worst case time, best case, average, space and complexities

4.Conclusion

Different sorting techniques have different uses according to their behavior for different inputs, every sorting technique has its own best case, average case and worst case according to inputs. Selection sort is useful where swapping is complex. Normally, Insertion sort is use for small data sets. For large data sets, Merge sort and Bubble sort are useful. Merge sort is practically useful for humans and as similar Bubble and Selection sort is used. For restricted data (data in a fixed interval) radix sort is useful. Bubble sort is rarely used, but found in teaching and theoretical expressions. We can use these all types of sorting to sort or arrange the array. These sorting techniques does not contain stack so it is quite easy to perform on array.

5.Acknowledge

I am very much thankful to my guide who gave me opportunity to share my ideas about Sorting in this paper. Sorting is the most important term used in computer science for sorting or storing the data into array. Most of the people is not sure about the sorting concept . So with the help of my paper they will definitely understand Sorting easily.

6.References

[1]. Dixit.J.B, Fundamentals of computers and programming in c, Laxmi Publication pvt.Ltd., second edition 2010.

[2]. http://en.wikipedia.org/wiki/Sorting_algorithm

[3]. http://en.wikipedia.org/wiki/Shellsort/Applications

[4]. http://www.cprogramming.com/tutorial/computersciencetheo ry/sortcomp.html [5]Seymour Lipschutz, Data Structures using C, McGraw-Hill, 2011

[5]. Ajay Kumar , Bharat Kumar , Chirag Dawar , Dinesh Bajaj, " Comparison Among Different Sorting Techniques" in International Journal For Research in Applied Science And Engineering, Technology (IJRAS ET)