

Name of Algorithm

1. Algorithm Introduction/ Description/Explanation

- Font will be **Times New Roman** as here.
- Font size should be **12**.
- Plagiarism must be **0%** in this section.
- Write in your own words not to copy from anywhere.
-

2. Flowchart

You can use following website for creating flowchart

<https://www.lucidchart.com/pages/examples/data-flow-diagram-software>

<https://creately.com/lp/data-flow-diagram-software-online/>

3. Algorithm

- Must be in **Courier New** Font and font size is **12** as mentioned in below sample example.

```
begin BubbleSort(list)

    for all elements of list
        if list[i] > list[i+1]
            swap(list[i], list[i+1])
        end if
    end for

    return list

end BubbleSort
```

4. Examples

Two examples with explanation, One with normal case and other one with either worst case or best case.

5. Complexity Analysis

The Sample of complexity analysis given as below for Merge sort without Plagiarism:

Mergesort function dividing a large list into small sublists recursively and at last, it calls merge function.

Here Merge function is comparing the elements of two sublists A, B and merge their elements by comparing them and save them into a newly created list. So it requires extra list of length n (length of list).

Recurrence Relation

Mergesort (Divide) \Rightarrow Divide the problem into two equal parts.

Merge (Conquer) \Rightarrow Compare ($\theta(n)$)

\Rightarrow Write ($\theta(n)$)

Final recurrence relation for merge sort

$$T(n) = 2T(n/2) + \theta(n)$$

By master's method

$$a = 2, b = 2, k = 1, p = 0$$

$$\text{Here } a = b^k \Rightarrow 2 = 2^1$$

Also $p > -1$,

$$T(n) = \theta(n^{\log_b a} * \log^{p+1} n)$$

$$T(n) = \theta(n^{\log_2 2} * \log^{0+1} n)$$

$$T(n) = \theta(n \log n)$$

Performance

Worst-case time complexity: $\theta(n \log n)$

Best-case time complexity: $\theta(n \log n)$

Average-case time complexity: $\theta(n \log n)$

Worst-case space complexity: $\theta(n)$ (extra list)

Advantages

For large data structures, merge sort is better because of its stable nature, unlike quick sort and heap sort. Its worst-case time complexity is $\theta(n \log n)$, which is better than any other algorithm.

Disadvantages

It requires $\theta(n)$ extra space for sorting. It's slower as compare to quick sort because of more function calling.

6. C and Python Implementation

- The C language and Python Implementation of the Algorithm is required here with all the required comments (//) and notations.
- Font Courier New and font size must be 12.

```
// A utility function to get maximum of two integers
int max(int a, int b)
{
    return (a > b)? a : b;
}
```

Note:

- A. The Start Writing the above Mentioned Points from Second Page.
- B. First Page with Contain Algorithm, and Participating Student Name and RollNo with their contribution as per following format.

Linear Search

ABC(R12345)- Introduction of Algorithm

DEF(R67890)-Flowchart and Algorithm

XYZ(R54321)-Complexity Analysis and Codes