System Software Lab

22XXXXX (ID)

May 2023

Contents

1 Introduction

A sorting algorithm is used to arrange elements of an array/list in a specific order. A sorting algorithm is considered stable if the two or more items with the same value maintain the same relative positions even after sorting.
There are various sorting algorithms that can be used to complete this operation. And, we can use any algorithm based on the requirement.

2 Types of Sorting algorithms

Sorting algorithms are a fundamental concept in computer science that arrange a collection of elements or data in a particular order. There are numerous sorting algorithms, each with its own characteristics, advantages, and disadvantages. Here are descriptions of some commonly used sorting algorithms:

1. Bubble Sort
2. Insertion Sort
3. Merge Sort
4. Quick Sort
5. Selection Sort
6. Counting Sort
7. Radix Sort
8. Heap Sort
9. Bucket Sort
10. Shell Sort

3 Complexity of Sorting Algorithms

The efficiency of any sorting algorithm is determined by the time complexity and space complexity of the algorithm.

- Time Complexity: Time complexity refers to the time taken by an algorithm to complete its execution with respect to the size of the input.

- Space Complexity: Space complexity refers to the total amount of memory used by the algorithm for a complete execution. It includes both the auxiliary memory and the input.

4 Time Complexity Equations

Time Complexity Equations for various Algorithms :

$$T(n) = 2T(n/2) + O(n) \tag{1}$$

$$T(n) = T(k) + T(n-k-1) + O(n) \tag{2}$$

$$T(n) = T(2n/3) + O(n) \tag{3}$$

equation 1 is the time complexity equation corresponding to Merge Sort and equation 2 is the time complexity equation corresponding to Quick Sort and equation 3 is the time complexity equation corresponding to Heap Sort.

5 Comparison

Let's see a complexity analysis of different sorting algorithms.

Sorting Algorithm	T.C (Best)	T.C (Avg)	T.C (Worst)	Space Complexity
Bubble Sort	n	n^2	n^2	1
Insertion Sort	n	n^2	$n2$	1
Merge Sort	nlogn	nlogn	nlogn	n
Quick Sort	nlogn	nlogn	$n2$	1
Selection Sort	$n2$	n^2	n^2	1
Counting Sort	n+k	n+k	n+k	max
Radix Sort	n+k	n+k	n+k	max

6 Conclusion

In conclusion, the choice of the best sorting algorithm depends on various factors such as the size of the data set, the degree of pre-sorting, stability requirements, and the trade-off between time complexity and space complexity. Here's a brief summary of the comparison and recommendations for different cases:

- Bubble Sort is simple to understand and implement, but it has a quadratic time complexity. It is suitable for small datasets or nearly sorted lists.
- Merge Sort has a time complexity of O(n log n) and is stable. It performs well on large datasets and is suitable for external sorting where data doesn't fit into memory.
- Quick Sort is efficient in practice. It is suitable for large datasets.