$O(n^2)$ $O(n)$? ? $O(\log n)$? $O(n \log n)$

Time Complexities ALGORITHMS

Sumit Soni

LINEAR SEARCH

✓ We simply traverse the list completely and match each element of the list with the item whose location is to be found.

Best Time Complexity	:	O(1)
Average Time Complexity	:	O(n)
Worst Time Complexity	:	O(n)

Sumit Soni

BINARY SEARCH

✓ In this approach, the element is always searched in the middle of a portion of an array. Binary search can be implemented only on a sorted list of items.

Best Time Complexity	:	O(1)
Average Time Complexity	:	O(logn)
Worst Time Complexity	:	O(logn)

Sumit Soni

BUBBLE SORT

✓ It is a sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order.

Best Time Complexity	:	$O(n)$
Average Time Complexity	:	$O(n^2)$
Worst Time Complexity	:	$O(n^2)$

Sumit Soni

SELECTION SORT

✓ The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning.

Best Time Complexity	:	$O(n^2)$
Average Time Complexity	:	$O(n^2)$
Worst Time Complexity	:	$O(n^2)$

Sumit Soni

INSERTION SORT

✓ Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

Best Time Complexity	:	$O(n)$
Average Time Complexity	:	$O(n^2)$
Worst Time Complexity	:	$O(n^2)$

Sumit Soni

MERGE SORT

✓ Merge sort is based on Divide and conquer method. It takes the list to be sorted and divide it in half to create two unsorted lists. The two unsorted lists are then sorted and merged to get a sorted list..

Best Time Complexity	:	O(nlogn)
Average Time Complexity	:	O(nlogn)
Worst Time Complexity	:	O(nlogn)

Sumit Soni

QUICK SORT

✓ Quick is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller array. A large array is partitioned into two arrays one of which holds values smaller than the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value.

Best Time Complexity	:	O(nlogn)
Average Time Complexity	:	O(nlogn)
Worst Time Complexity	:	$O(n^2)$

Sumit Soni

HEAP SORT

✓ Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the minimum element and place the minimum element at the beginning. We repeat the same process for the remaining elements.

Best Time Complexity	:	O(nlogn)
Average Time Complexity	:	O(nlogn)
Worst Time Complexity	:	O(nlogn)

Sumit Soni

Bucket SORT

✓ Bucket sort, or bin sort, is a sorting algorithm that works by distributing the elements of an array into a number of buckets. Each bucket is then sorted individually, either using a different sorting algorithm, or by recursively applying the bucket sorting algorithm.

Best Time Complexity	:	$O(n+k)$
Average Time Complexity	:	$O(n+k)$
Worst Time Complexity	:	$O(n^2)$

Sumit Soni

Radix SORT

✓ The idea of Radix Sort is to do digit by digit sort starting from least significant digit to most significant digit. Radix sort uses counting sort as a subroutine to sort.

Best Time Complexity	:	O(nk)
Average Time Complexity	:	O(nk)
Worst Time Complexity	:	O(nk)

Sumit Soni

DID YOU FIND THIS POST USEFUL?

Let me know in the Comments below!