

# Advantages & Disadvantages of Sorting Algorithms

## Selection Sort

Advantages	Disadvantages
The main advantage of the selection sort is that it performs well on a small list.	The primary disadvantage of the selection sort is its poor efficiency when dealing with a huge list of items.
Because it is an in-place sorting algorithm, no additional temporary storage is required beyond what is needed to hold the original list.	The selection sort requires $n^2$ number of steps for sorting $n$ elements.
Its performance is easily influenced by the initial ordering of the items before the sorting process.	Quick Sort is much more efficient than selection sort

# Insertion Sort

Advantages	Disadvantages
The main advantage of the insertion sort is its simplicity.	The disadvantage of the insertion sort is that it does not perform as well as other, better sorting algorithms
It also exhibits a good performance when dealing with a small list.	With $n^2$ steps required for every $n$ element to be sorted, the insertion sort does not deal well with a huge list.
The insertion sort is an in-place sorting algorithm so the space requirement is minimal.	The insertion sort is particularly useful only when sorting a list of few items.

# Bubble Sort

Advantages	Disadvantages
The primary advantage of the bubble sort is that it is popular and easy to implement.	The main disadvantage of the bubble sort is the fact that it does not deal well with a list containing a huge number of items.
In the bubble sort, elements are swapped in place without using additional temporary storage.	The bubble sort requires $n^2$ processing steps for every $n$ number of elements to be sorted.
The space requirement is at a minimum	The bubble sort is mostly suitable for academic teaching but not for real-life applications.

# Quick Sort

Advantages	Disadvantages
The quick sort is regarded as the best sorting algorithm.	The slight disadvantage of quick sort is that its worst-case performance is similar to average performances of the bubble, insertion or selections sorts.
It is able to deal well with a huge list of items.	If the list is already sorted than bubble sort is much more efficient than quick sort
Because it sorts in place, no additional storage is required as well	If the sorting element is integers than radix sort is more efficient than quick sort.

# Merge Sort

Advantages	Disadvantages
It can be applied to files of any size.	Requires extra space $\approx N$
Reading of the input during the run-creation step is sequential ==> Not much seeking.	Merge Sort requires more space than other sort.
If heap sort is used for the in-memory part of the merge, its operation can be overlapped with I/O	Merge sort is less efficient than other sort

# Binary Search Tree

Advantages	Disadvantages
Binary search tree is fast in insertion and deletion operations etc when balanced.	The shape of the binary search tree totally depends on the order of insertions and it can be regenerated, for the same key elements ,we can get different shape bst.
Binary search tree is very efficient and its code is easier When compared to linked list code.	It takes a long time to search an element in a bst because key value of each node has to be compared With the key element to be searched.
Duplicate keys or elements not allowed in binary search Tree.	Height of binary search tree is not under control because on depends on how elements are inserted

Algorithm	Time Complexity		
	Best	Average	Worst
Selection Sort	$\Omega(n^2)$	$\theta(n^2)$	$O(n^2)$
Bubble Sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$
Insertion Sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$
Quick sort	$\Omega(n \log(n))$	$\theta(n \log(n))$	$O(n^2)$
Merge sort	$\Omega(n \log(n))$	$\theta(n \log(n))$	$O(n \log(n))$