WELCOME

Comparative Study on Different Algorithms

M Shahriar Ishtiaque

191-15-12938

What is Sorting?

Sorting is a process of arranging a set of data or numbers in a Ascending Order or Descending Order.

Example:

Sort the numbers given below in Descending order

10, 30,20,50,60,40,100

Sorted: 100,60,50,40,30,20,10

Sorting is done by various algorithms to manage the time complexity of problems.

There are various kind of sorting algorithms.

Like –

- **Bubble Sort**.
- **Recursive Bubble Sort**.
- **Insertion Sort**.
- **Recursive Insertion Sort**.
- **Merge Sort**.
- **Iterative Merge Sort**.
- **Quick Sort**

Now I will be discussing about :

Insertion Sort.

Merge Sort.

Quick Sort.

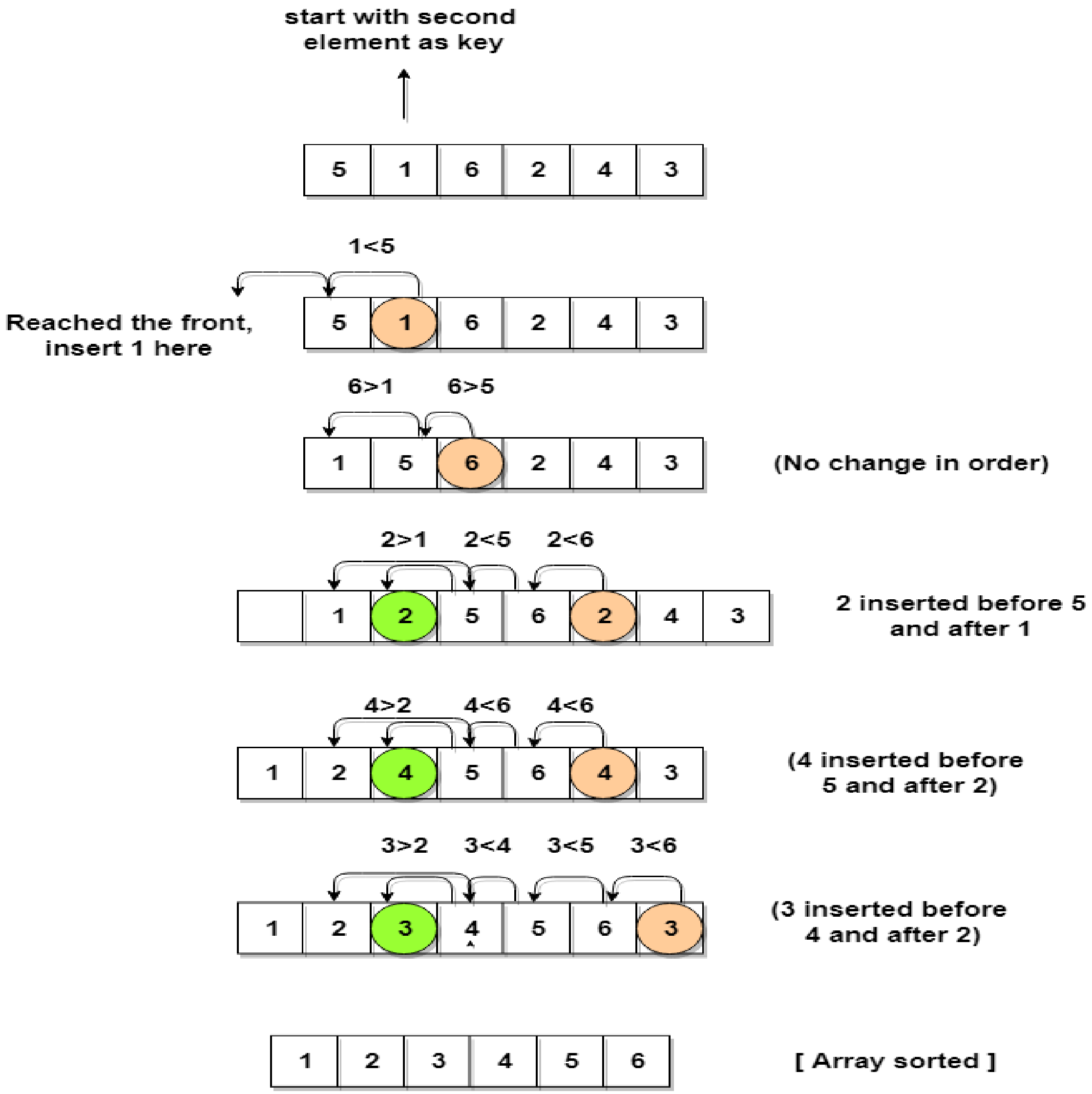
INSERTION SORT

One element from the input elements is consumed in each iteration to find it’s correct position, the position to which it belongs in sorted array.

If the current element is greater than the elements of it’s left side than leave it. Else, shift all the elements which is larger than the current elements.

Time Complexity:

Worst complexity: n^2
Average complexity: n^2
Best complexity: n
Space complexity: 1



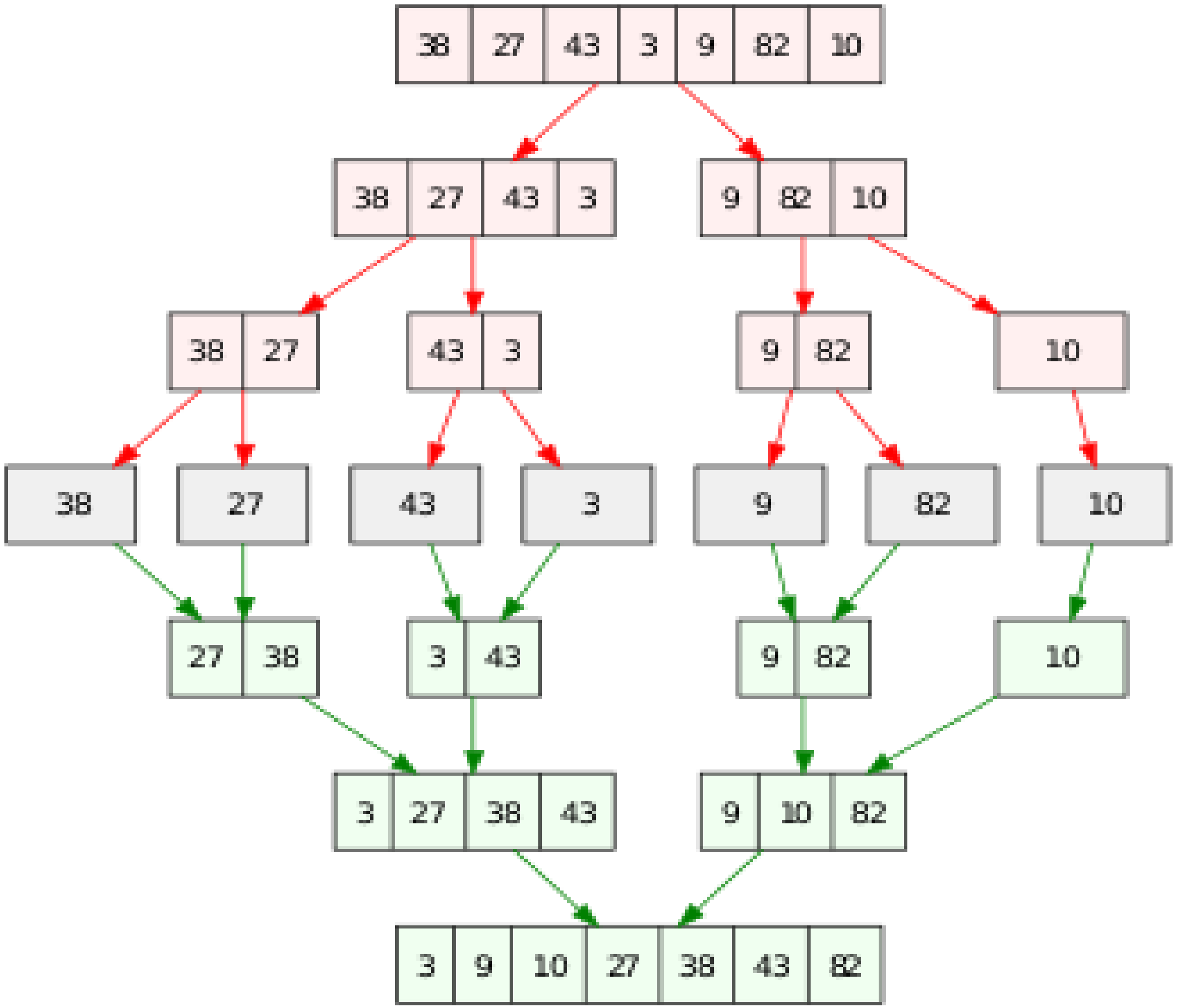
Merge Sort

Merge sort is a sorting technique based on divide and conquer technique. Merge sort first divides the array into equal halves and then combines them in a sorted manner.

Merge Sort Example:

Time Complexity

Worst complexity: n*log(n)
Average complexity: n*log(n)
Best complexity: n*log(n)



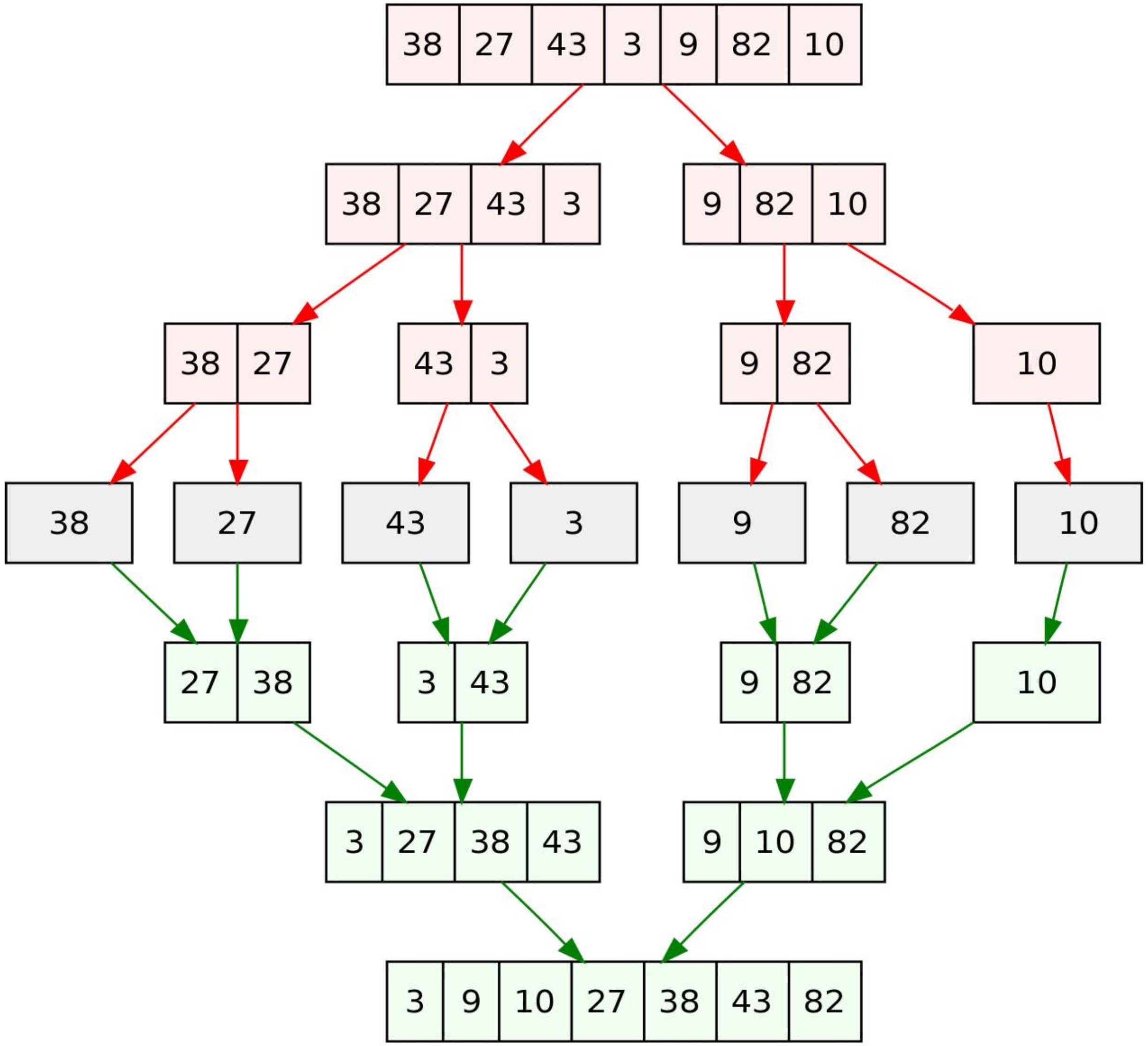
Quick Sort

Quicksort is a divide-and-conquer algorithm. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then sorted recursively.

Quick Sort example:

Time complexity

Worst case complexity: n^2
Average case complexity: n*log(n)
Best case complexity: n*log(n)



Difference between Quick Sort and Insertion Sort:

BASIS FOR COMPARISON	QUICK SORT	INSERTION SORT
Efficiency	More efficient	Less efficient
Speed	Faster	Slower
Worst case complexity	O(n^2)	n^2
Best case complexity	n*log(n)	n

Difference between Quick Sort and Merge Sort:

BASIS FOR COMPARISON	QUICK SORT	MERGE SORT
Partitioning of the elements in the array	The splitting of a list of elements is not necessarily divided into half.	Array is always divided into half (n/2).
Worst case complexity	$O(n^2)$	O(n log n)
Works well on	Smaller array	Operates fine in any type of array.
Speed	Faster than other sorting algorithms for small data set.	Consistent speed in all type of data sets.
Additional storage space requirement	Less	More
Efficiency	Inefficient for larger arrays.	More efficient.
Sorting method	Internal	External

Thank you