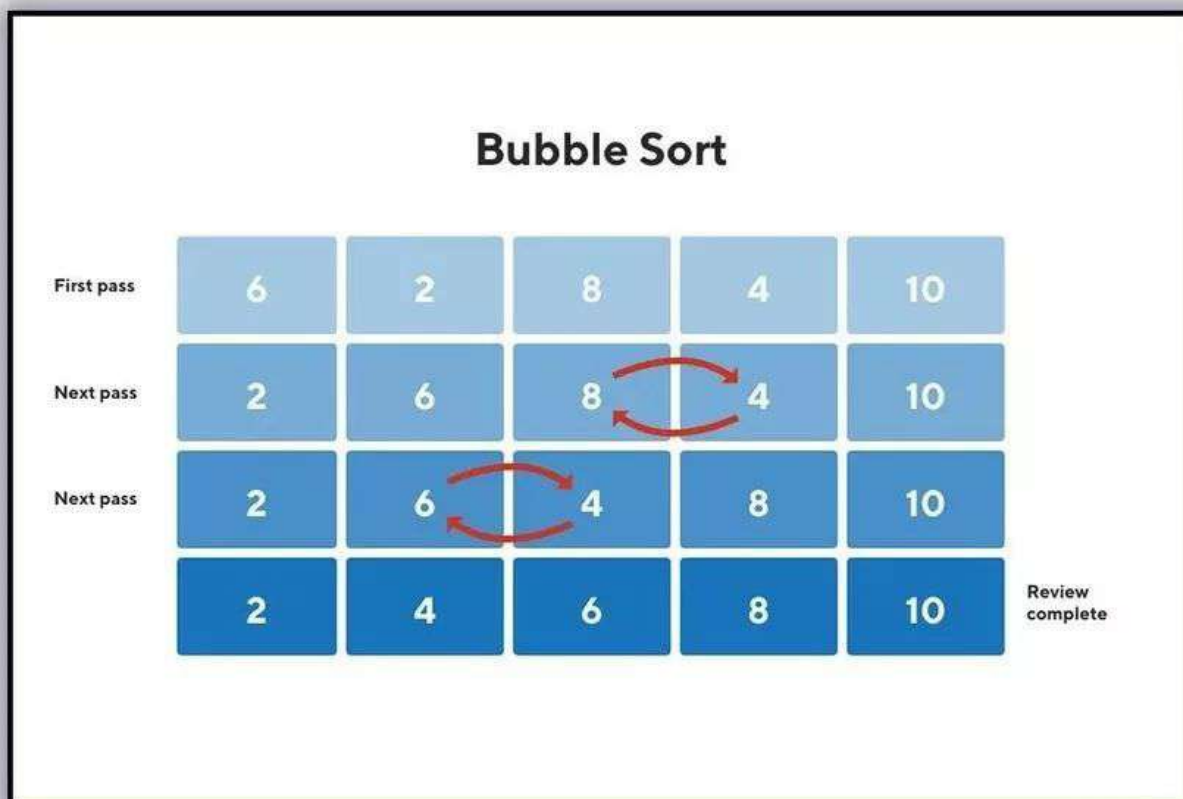# Sorting
Algorithms

Bubble Sort

Bubble sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexities is quite high.

Bubble Sort

|					
First pass	6	2	8	4	10
Next pass	2	6	8	4	10
Next pass	2	6	4	8	10
	2	4	6	8	10	Review complete

Heap Sort

Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to the selection sort where we first find the minimum element and place the minimum element at the beginning. Repeat the same process for remaining elements.
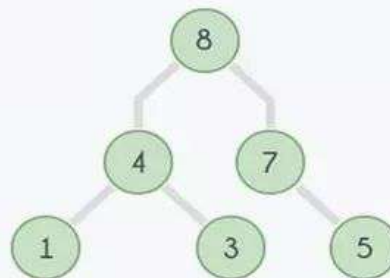
Arr		4	3	7	1	8	5
	0	1	2	3	4	5	6

Initial Elements

```
        4
       / \
      3   7
     /   / \
    1   8   5
```

Max Heap

```
        8
       / \
      4   7
     /   / \
    1   3   5
```
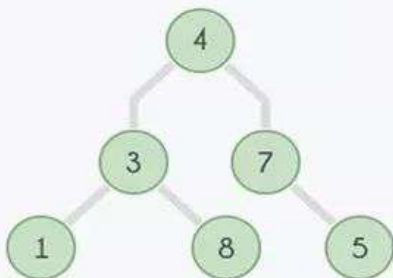
Insertion Sort

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands The array is virtualy 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.

|									
54	26	93	17	77	31	44	55	20	Assume 54 is a sorted list of 1 item
26	54	93	17	77	31	44	55	20	inserted 26
26	54	93	17	77	31	44	55	20	inserted 93
17	26	54	93	77	31	44	55	20	inserted 17
17	26	54	77	93	31	44	55	20	inserted 77
17	26	31	54	77	93	44	55	20	inserted 31
17	26	31	44	54	77	93	55	20	inserted 44
17	26	31	44	54	55	77	93	20	inserted 55
17	20	26	31	44	54	55	77	93	inserted 20

Merge Sort

Merge Sort is a algorithm that is considered an example of the divide and conquer strategy. So, in this algorithm the array is initially divided into two equal halves and then they are combine in a sorted manner.
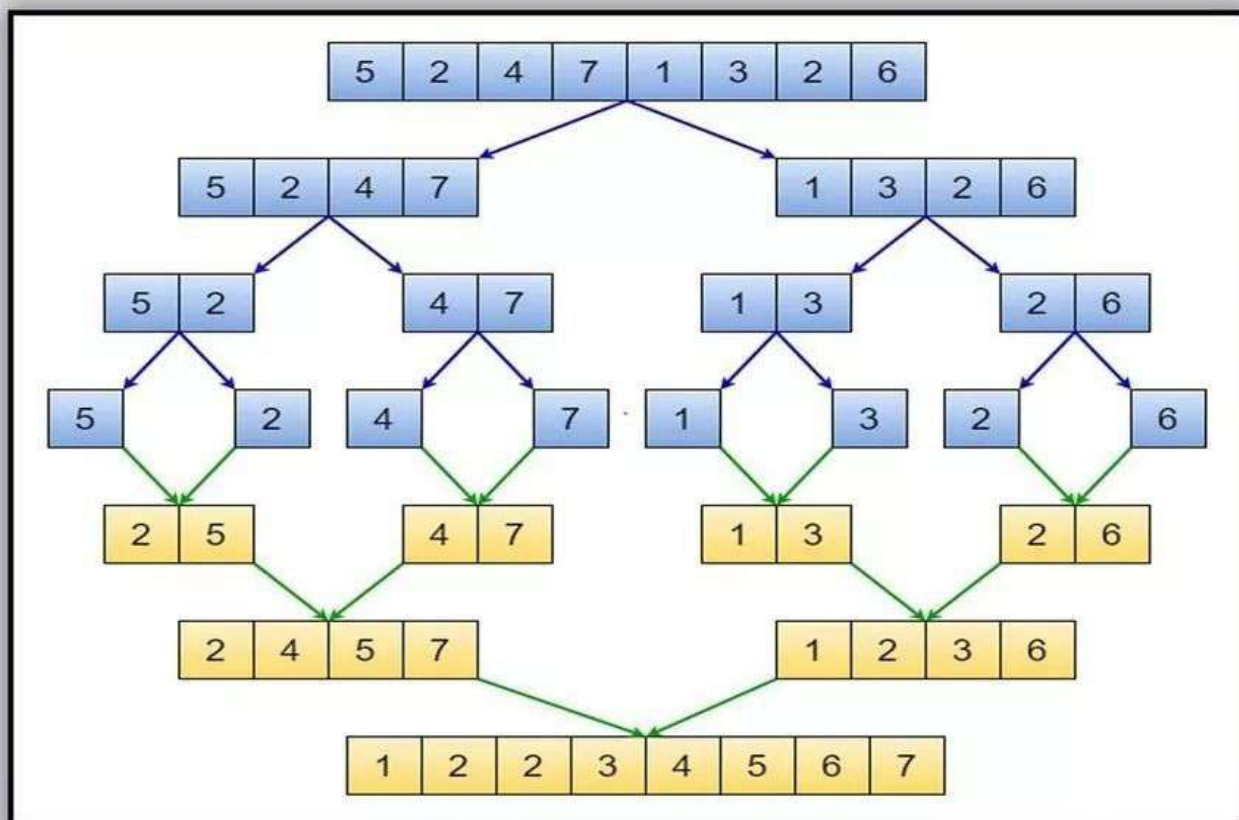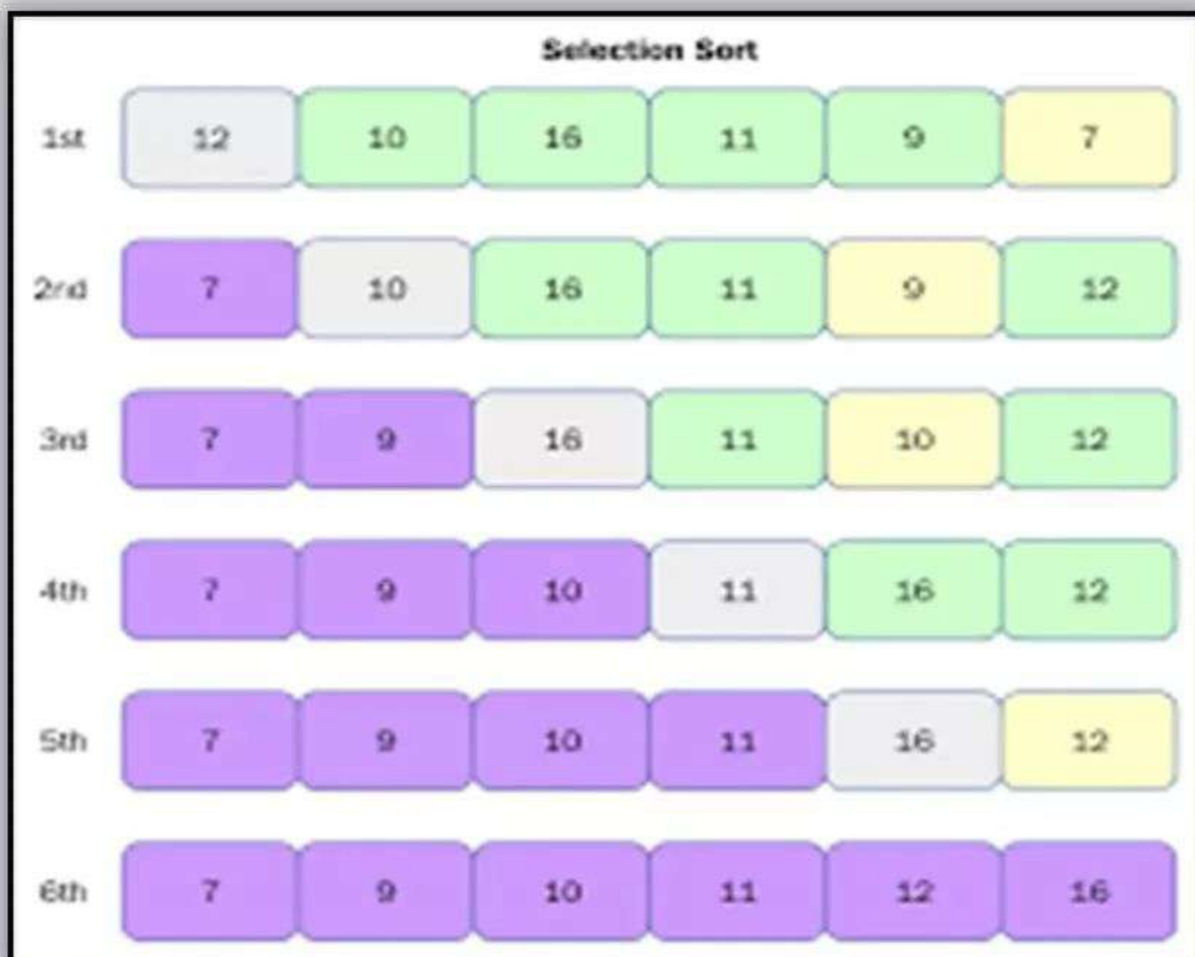
Quick Sort

Quick sort is also an divide and conquer algorithm. It picks an element as a pivot and partitions the given array around the picked pivot. This algorithm is used to quickly sort items within an array no matter how big the array is.
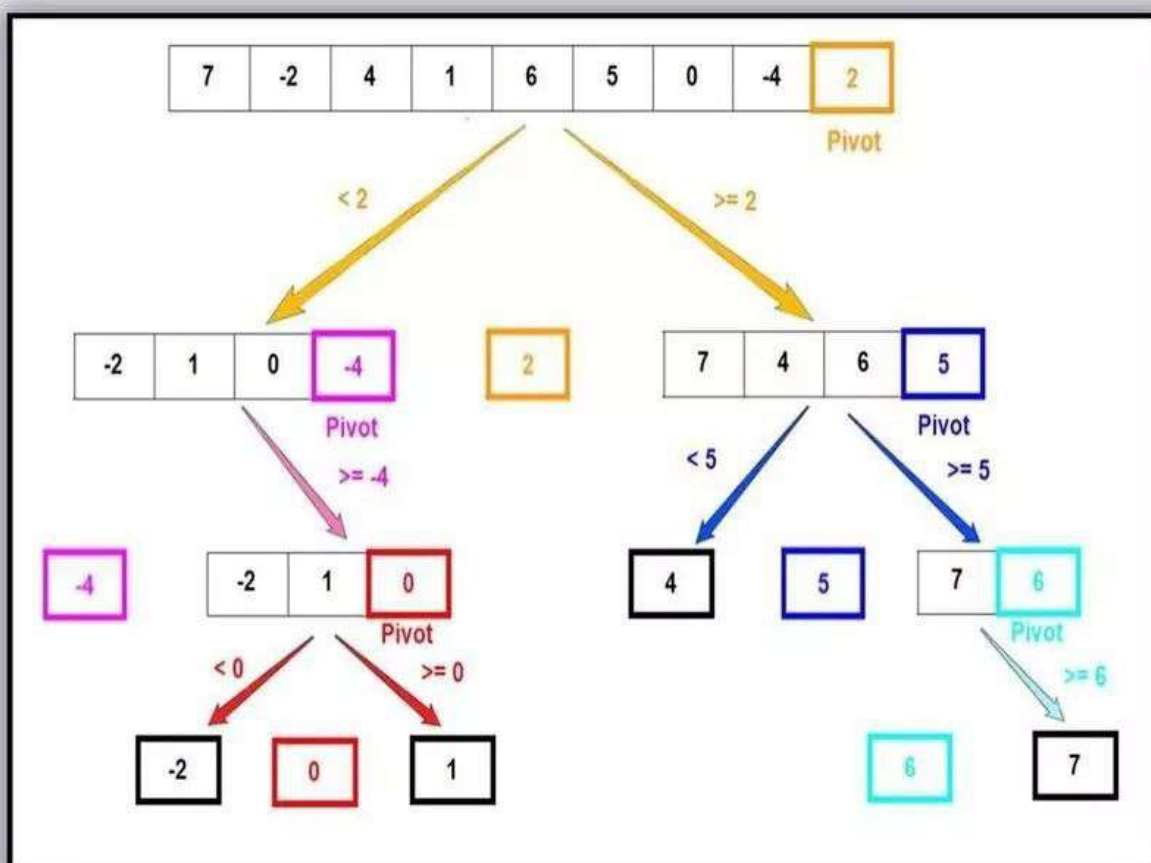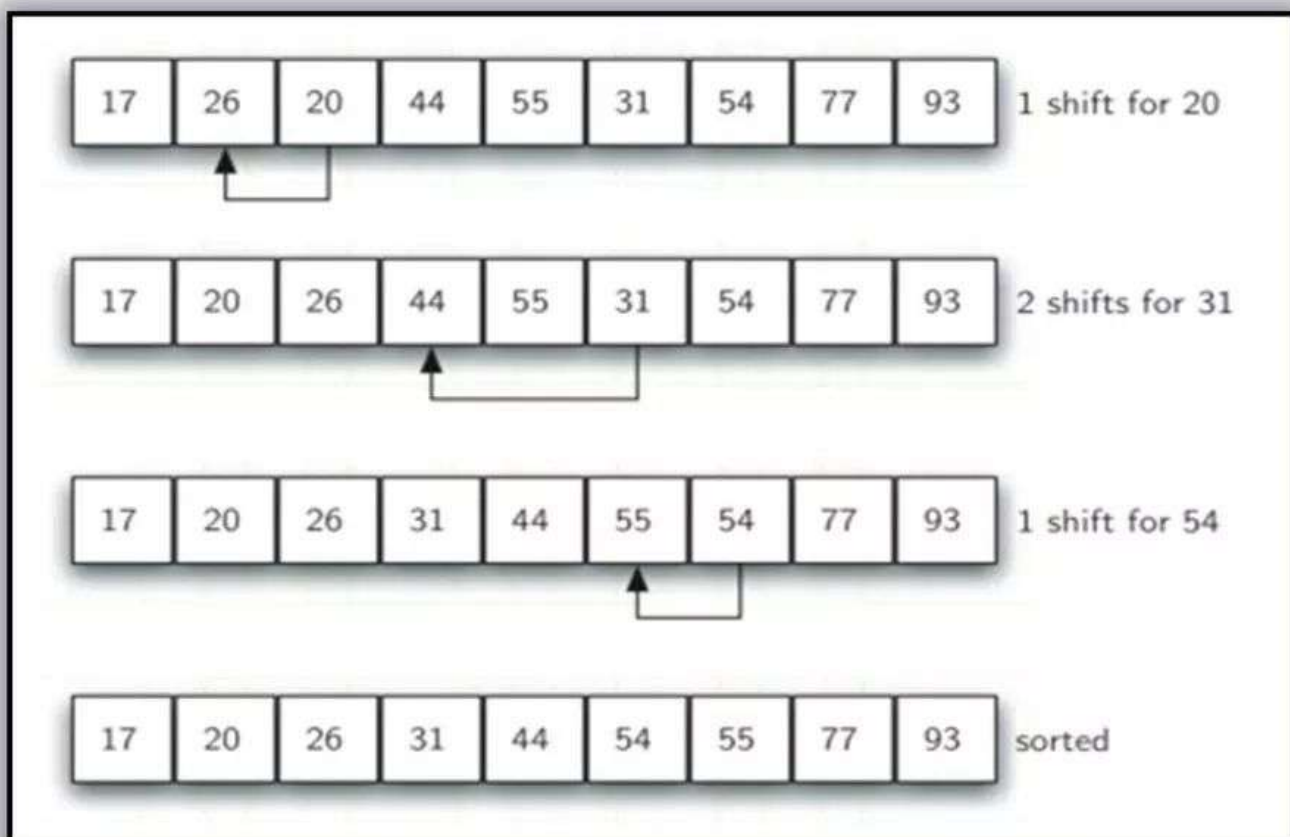
Selection Sort

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.

Selection Sort

1st	12	10	16	11	9	7
2nd	7	10	16	11	9	12
3rd	7	9	16	11	10	12
4th	7	9	10	11	16	12
5th	7	9	10	11	16	12
6th	7	9	10	11	12	16

Shell Sort

Shell sort is mainly a variation of Insertion Sort. In insertion sort, we move elements only one position ahead. When an element has to be moved far ahead, many movements are involved. The idea of ShellSort is to allow the exchange of far items. In Shell sort, we make the array h-sorted for a large value of h. We keep reducing the value of h until it becomes 1. An array is said to be h-sorted if all sublists of every h'th element are sorted.

| 17 | 26 | 20 | 44 | 55 | 31 | 54 | 77 | 93 | 1 shift for 20 |

| 17 | 20 | 26 | 44 | 55 | 31 | 54 | 77 | 93 | 2 shifts for 31 |

| 17 | 20 | 26 | 31 | 44 | 55 | 54 | 77 | 93 | 1 shift for 54 |

| 17 | 20 | 26 | 31 | 44 | 54 | 55 | 77 | 93 | sorted |

Time Complexities

Sorting Algorithm	Best Case	Average Case	Worst Case
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$
Heap Sort	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$
Merge Sort	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$
Quick Sort	$O(n\log n)$	$O(n\log n)$	$O(n^2)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Shell Sort	$O(n(\log n)^2)$	$O(n^{3/2})$	$O(n)$