**Bubble Sort Algorithm:**

Bubble Sort, is a simple sorting algorithm which sorts n number of element’s in the list by comparing the each pair of adjacent items and swaps them if they are in wrong order.

**Worst and Average Case Time Complexity:**O(n\*n).

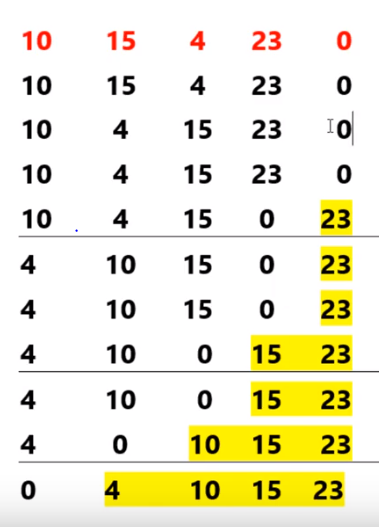
**Best Case Time Complexity:** O(n).

**Steps:**

1. Starting with the first element (index=0) compare the current element with the next element of the list.

2. I f the current element is greater than the next element of the list, swap them.

3. If the Current element is less than the next element, move to the next element. Repeat step1.



**Selection Sort algorithm:**

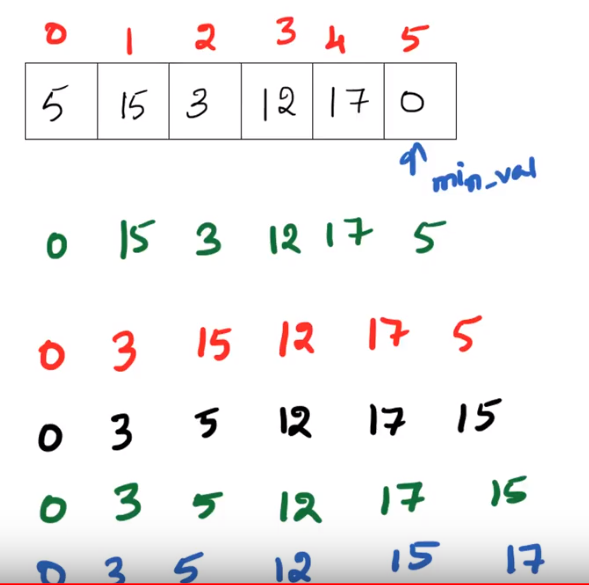
Selection sort is an in placed comparison based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end.

**Steps:**

1. Starting from the first element search for the smallest (biggest) element in the list of numbers.

2. Swap minimum (or maximum) number with the first element.

3. Take the sub list(ignore sorted part) and repeat step 1 and 2 until all the elements are sorted.



**Quick sort Algorithm:**

Quicksort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.

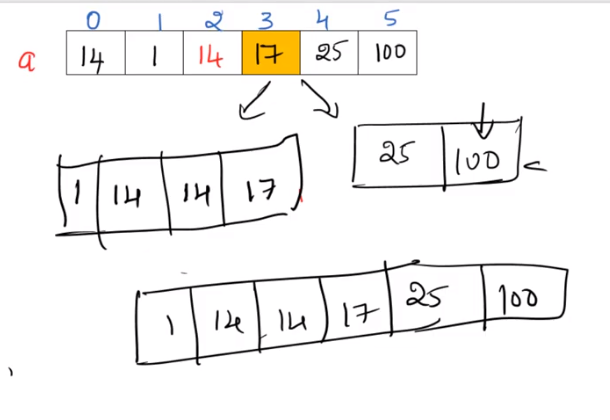
**Time complexity of Quicksort** is O(n2)

1. Always pick first element as pivot.
2. Always pick last element as pivot
3. Pick a random element as pivot.
4. Pick median as pivot.

**Steps:**

* 1. Select the pivot elemenet.
  2. Find the correct position of pivot elemet in the list by rearranging it.
  3. Divide the list based on pivot element.
  4. Sort the sublist recursively.

First element as pivot element:

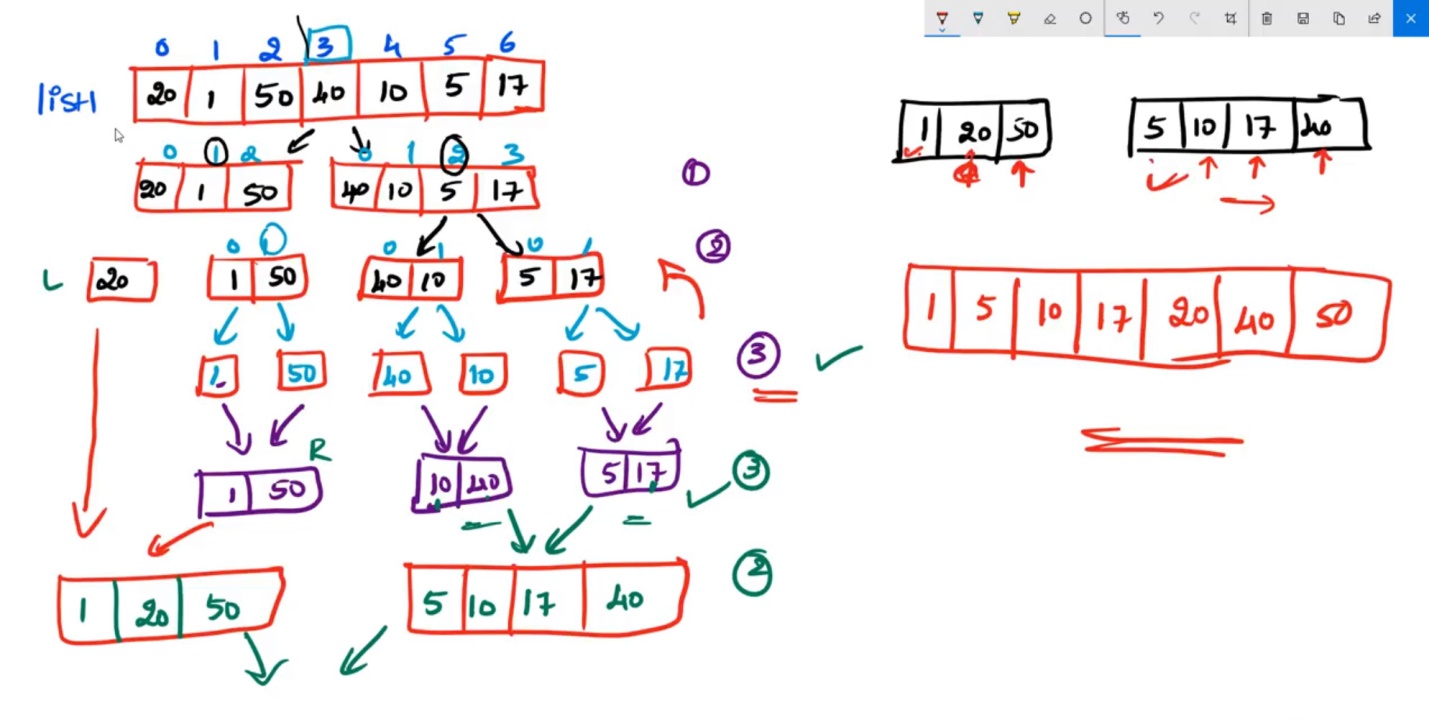


**Merge sort**

Like Quicksort, Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves.

**Time complexity of Merge sort** is O(nLogn).

1. Split the unsorted list.
2. Compare each of the elements and group them.
3. Repeat step2 until whole list is merged and sorted



| **Algorithm** | **Time Complexity** | | |
| --- | --- | --- | --- |
|  | **Best** | **Average** | **Worst** |  |
| [Selection Sort](http://geeksquiz.com/selection-sort/) | Ω(n^2) | θ(n^2) | O(n^2) |  |
| [Bubble Sort](http://geeksquiz.com/bubble-sort/) | Ω(n) | θ(n^2) | O(n^2) |  |
| [Quick Sort](http://geeksquiz.com/quick-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n^2) |  |
| [Merge Sort](http://geeksquiz.com/merge-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) |  |