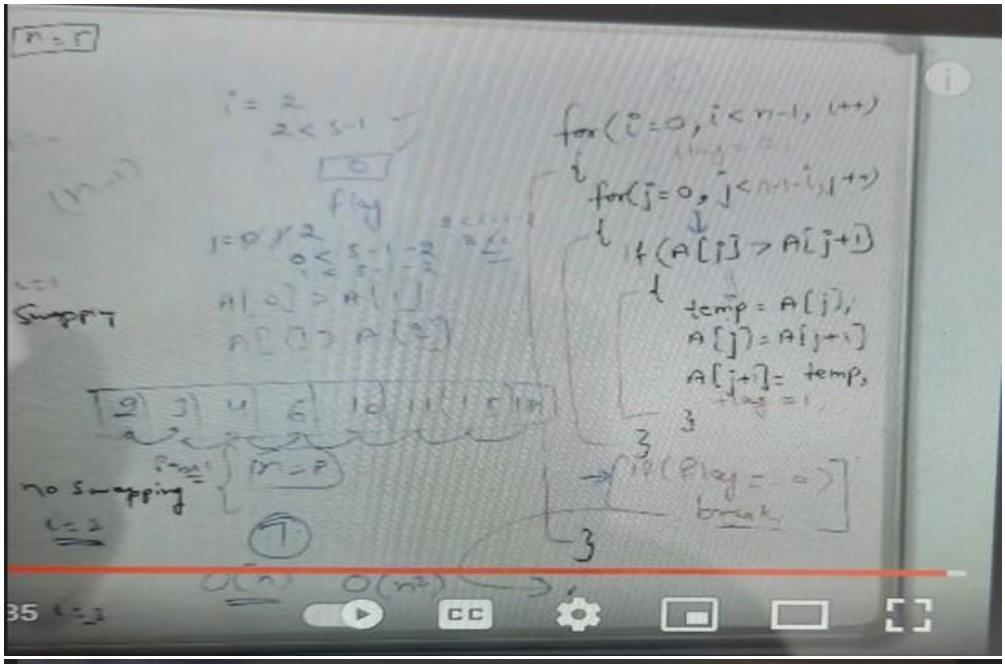
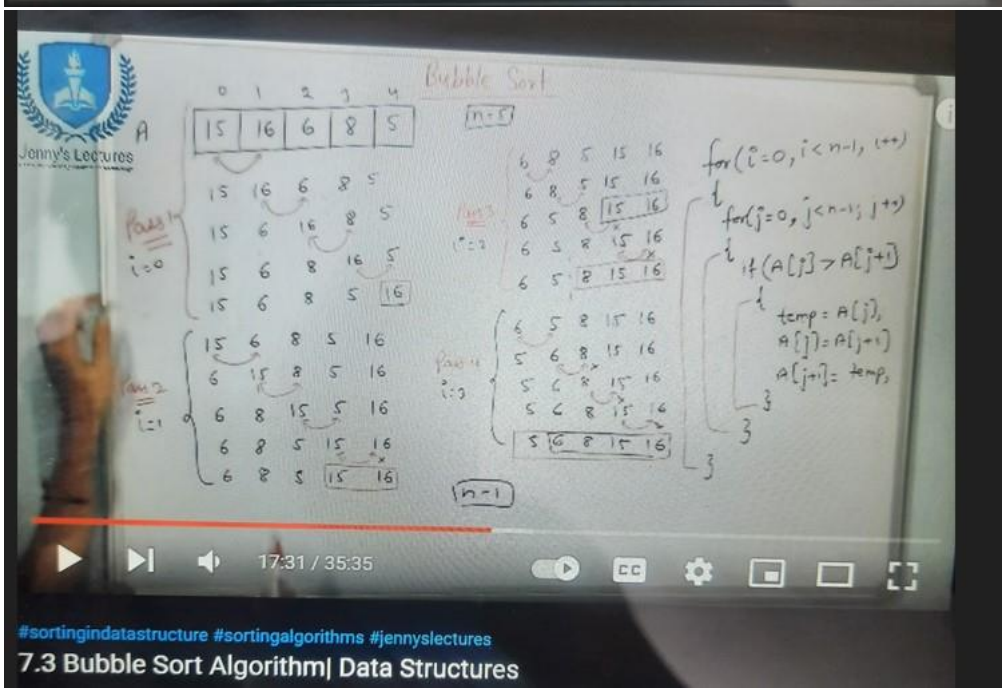
Algorithm Overview:

Bubble sort:

https://www.youtube.com/watch?v=o4bAoo_gFBU&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=97

https://github.com/SushantPoman/DataStructures/blob/master/src/BubbleSort.java



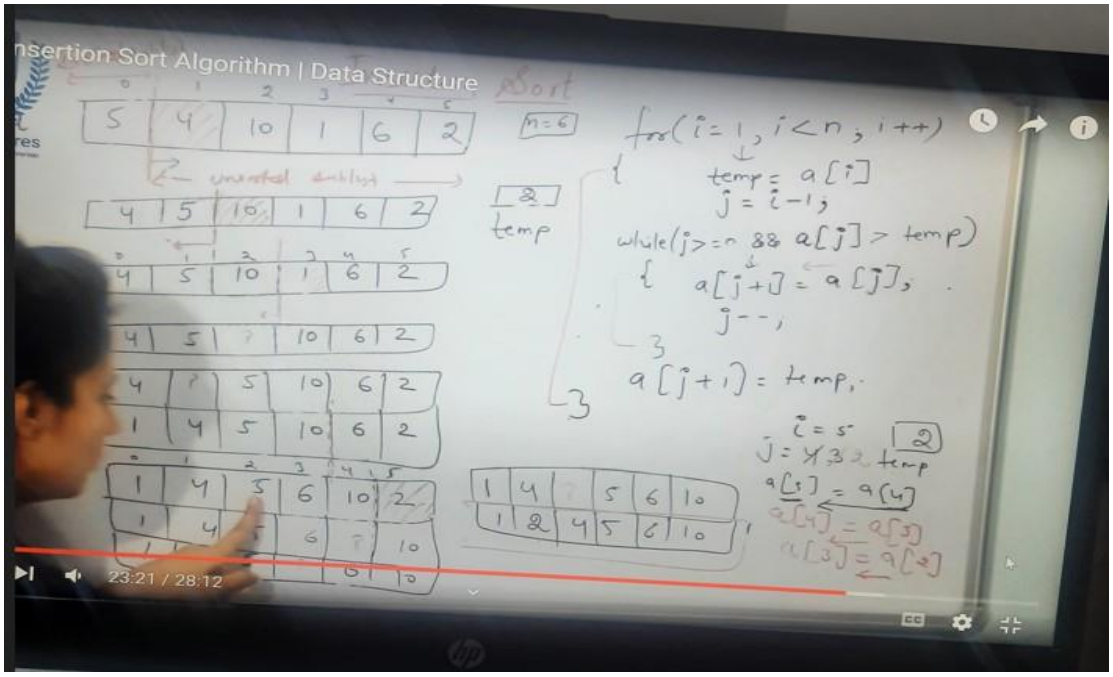


Insertion sort:

https://www.youtube.com/watch?v=yCxV0kBpA6M&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=99

https://github.com/SushantPoman/DataStructures/blob/master/src/InsertionSort.java

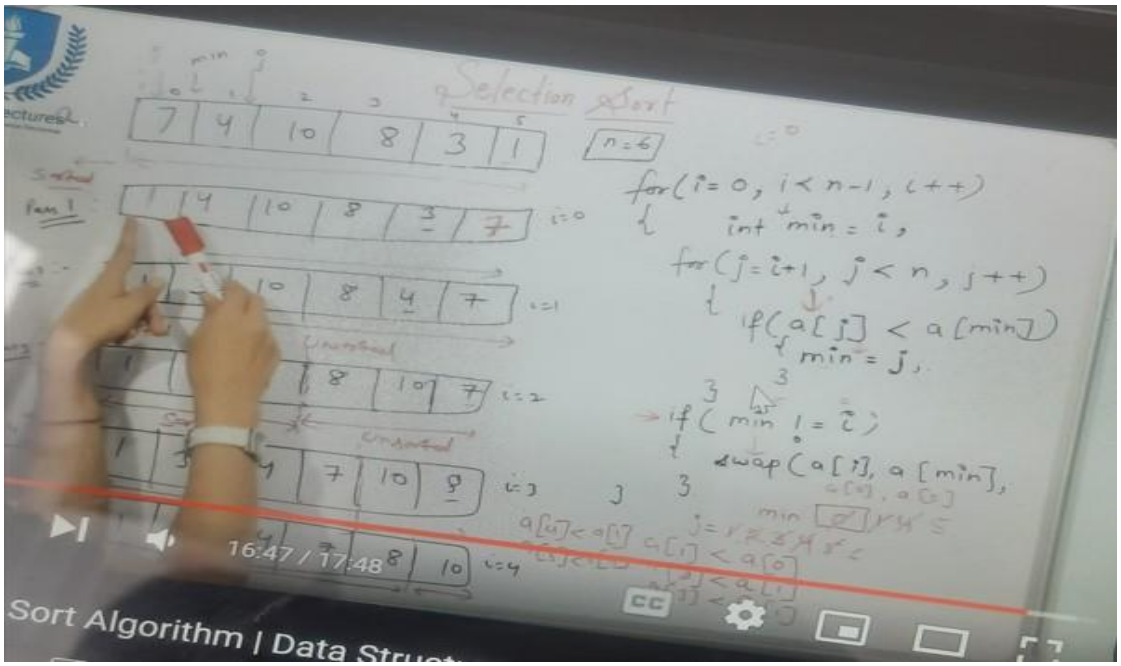
Given array is divided into two parts. Sorted array and Unsorted array.



Selection sort:

https://www.youtube.com/watch?v=9oWd4VJOwr0&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=100

https://github.com/SushantPoman/DataStructures/blob/master/src/SelectionSort.java



Quicksort:

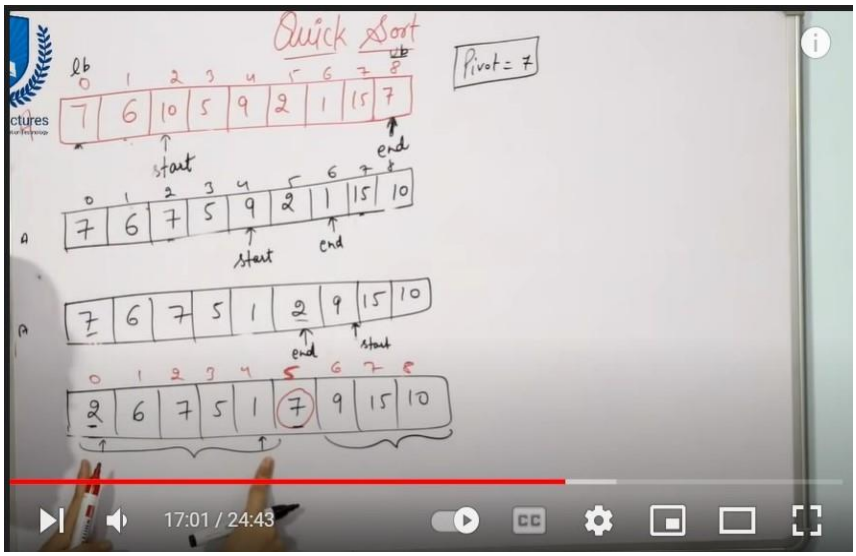
https://www.youtube.com/watch?v=QN9hnmAgmOc&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=100

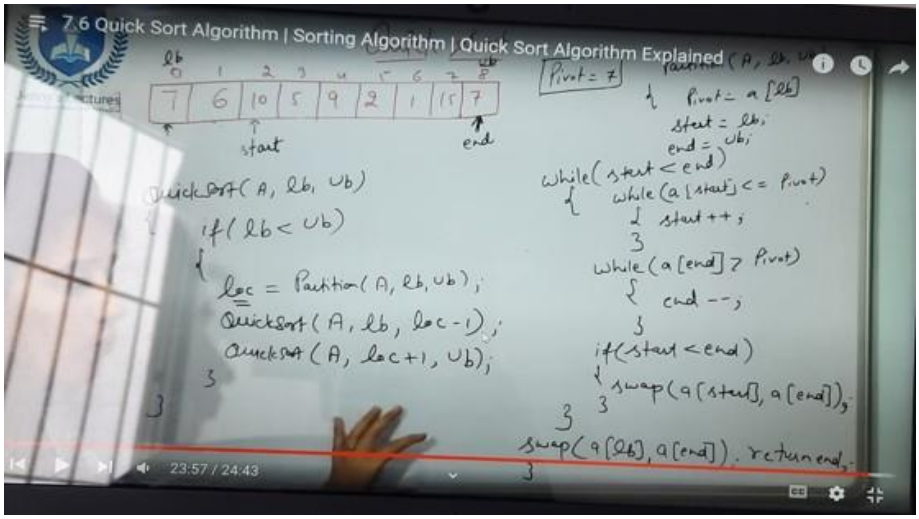
https://github.com/SushantPoman/DataStructures/blob/master/src/QuickSort.java

Quicksort algorithm is based on the divide and conquer approach where an array is divided into subarrays by selecting a pivot element.

While dividing the array, the pivot element should be positioned in such a way that elements less than pivot are kept on the left side and elements greater than pivot are on the right side.

The same process is continued for both left and right subarrays. Finally, sorted elements are combined to form a sorted array.

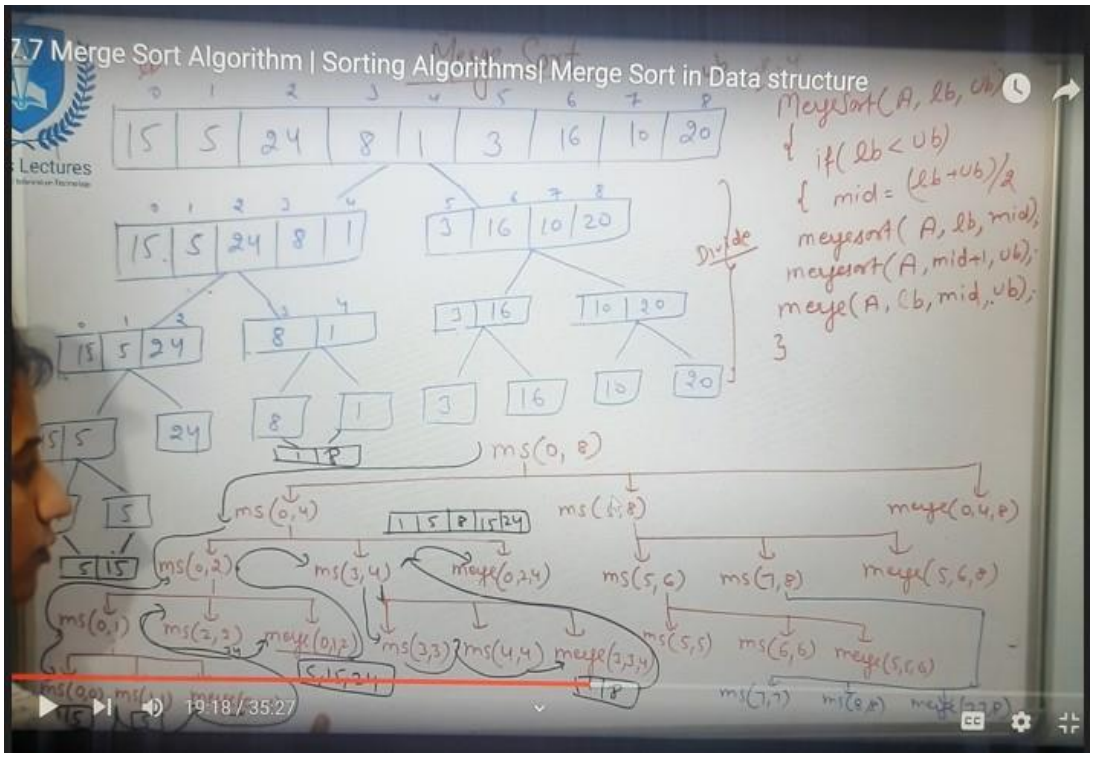


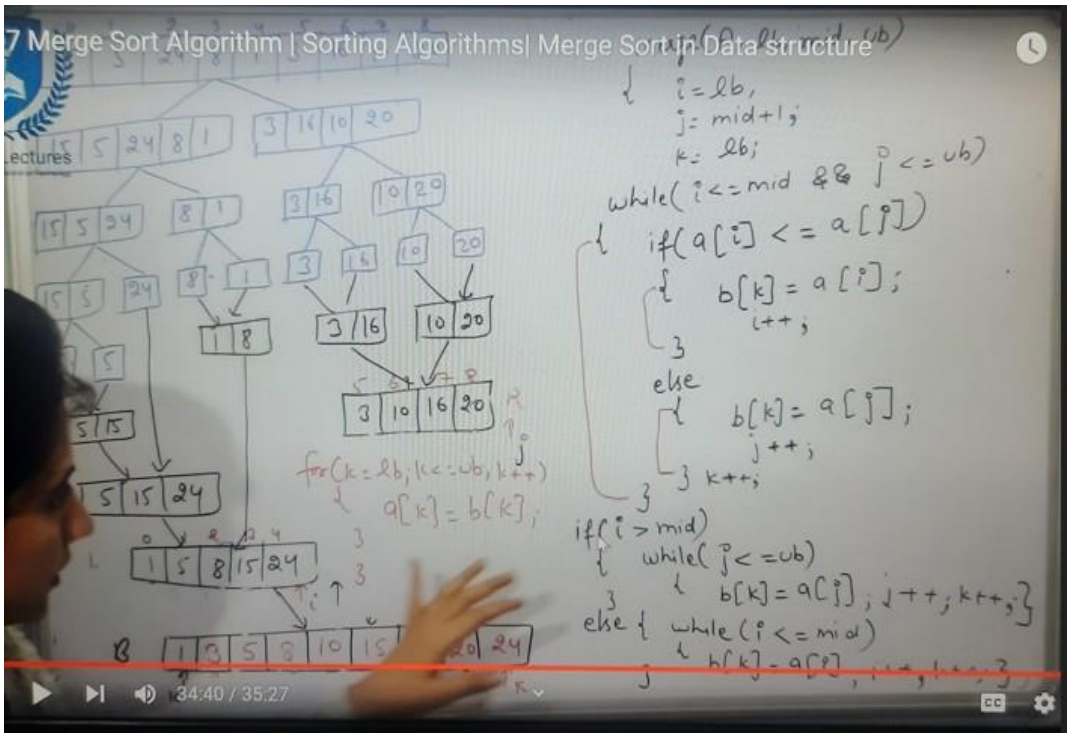


Merge sort:

https://www.youtube.com/watch?v=jlHkDBEumP0&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&iĀĀĀĀĀĀndex=101

https://github.com/SushantPoman/DataStructures/blob/master/src/MergeSort.java

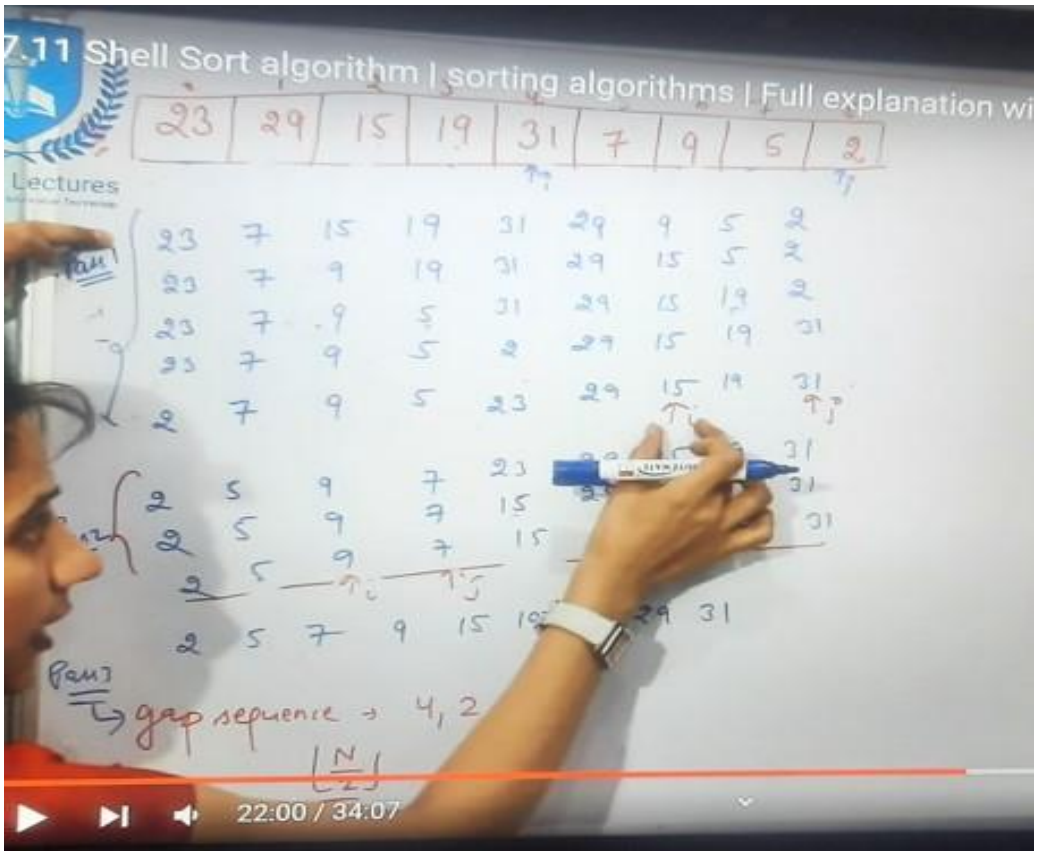


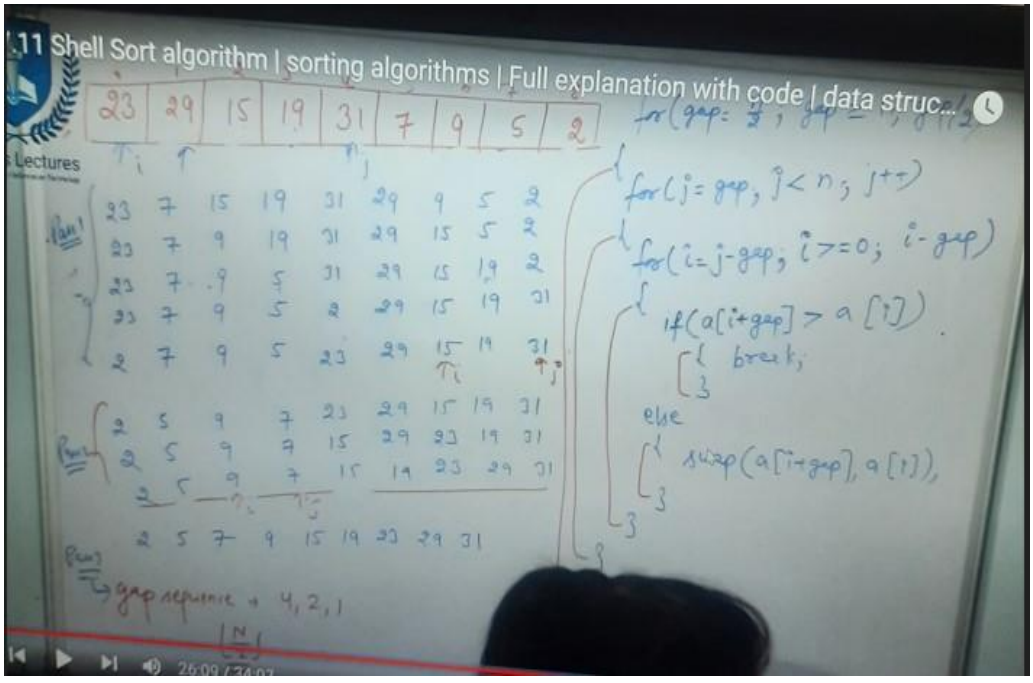


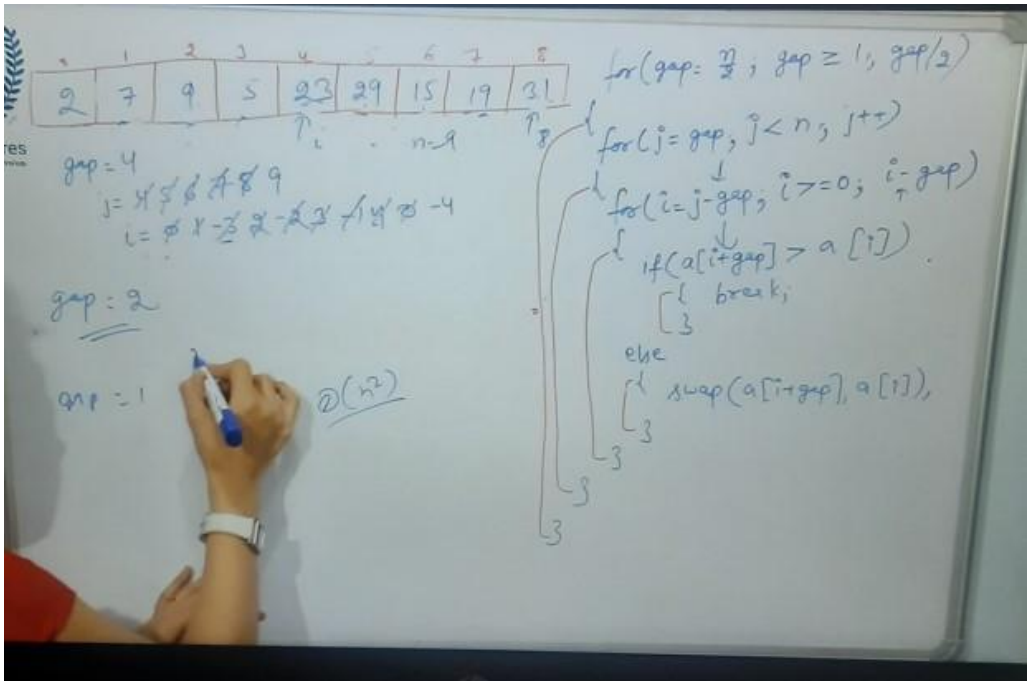
Shell sort:

https://www.youtube.com/watch?v=9crZRd8GPWM&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=105

https://github.com/SushantPoman/DataStructures/blob/master/src/ShellSort.java



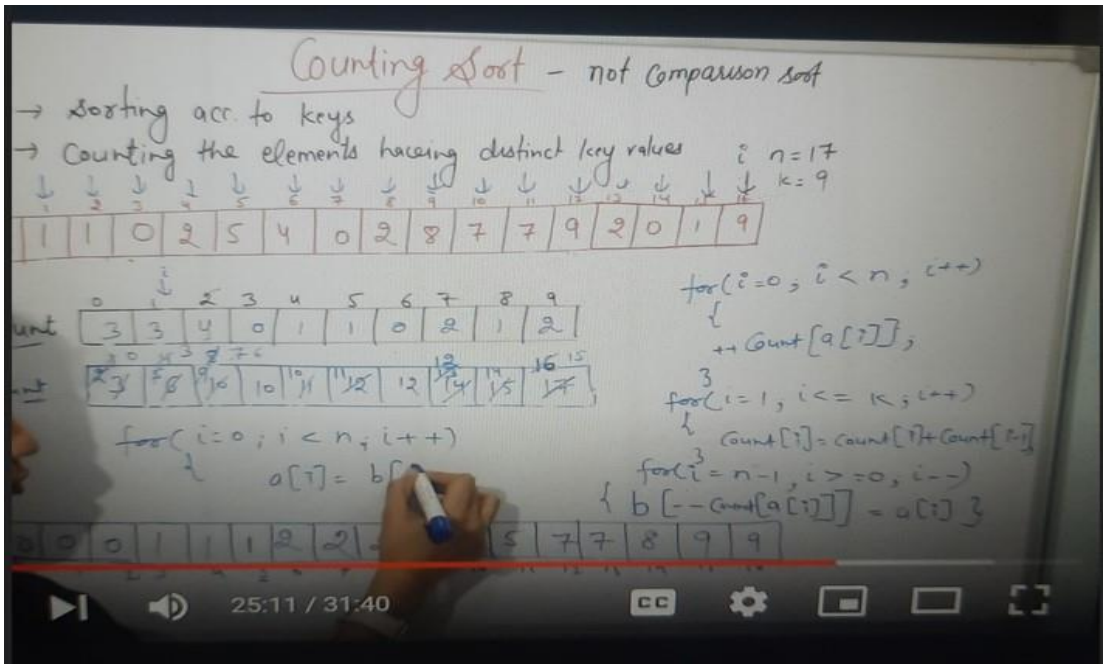


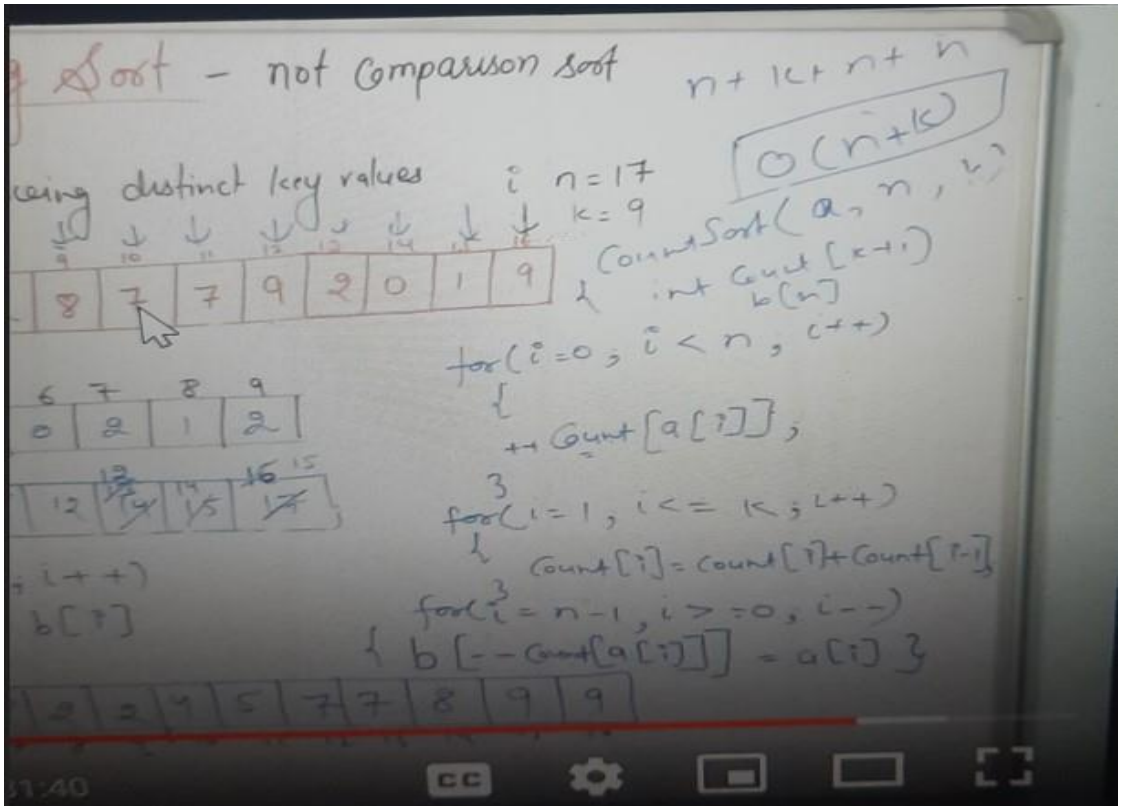

0 1 2 3 4 5 6 7 8
2 7 9 5 23 29 15 19 31
n-1
gap = 4
gap : 2
gap : 1
O(n²)
for(gap = n/2 ; gap >= 1; gap/2)
for(j = gap, j < n; j++)
for(i = j - gap; i >= 0; i - gap)
if(a[i+gap] > a[i])
break;
else
swap(a[i+gap], a[i]),

Counting sort:

https://www.youtube.com/watch?v=pEJiGC-ObQE&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=106

https://github.com/SushantPoman/DataStructures/blob/master/src/CountSort.java

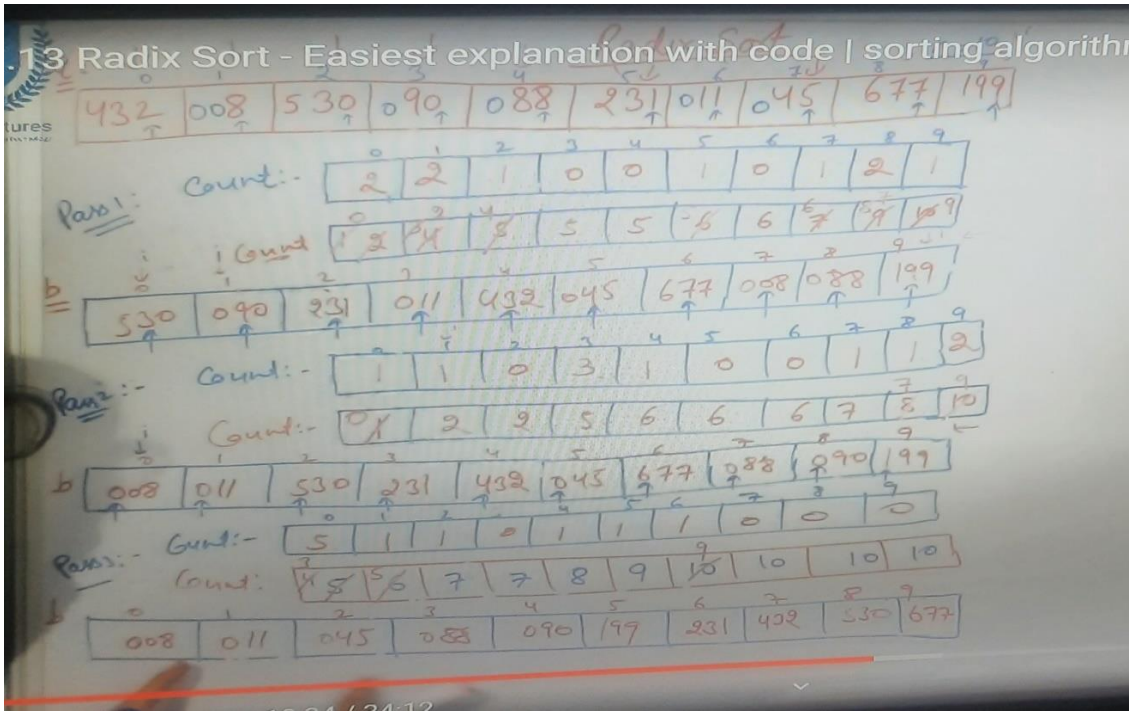


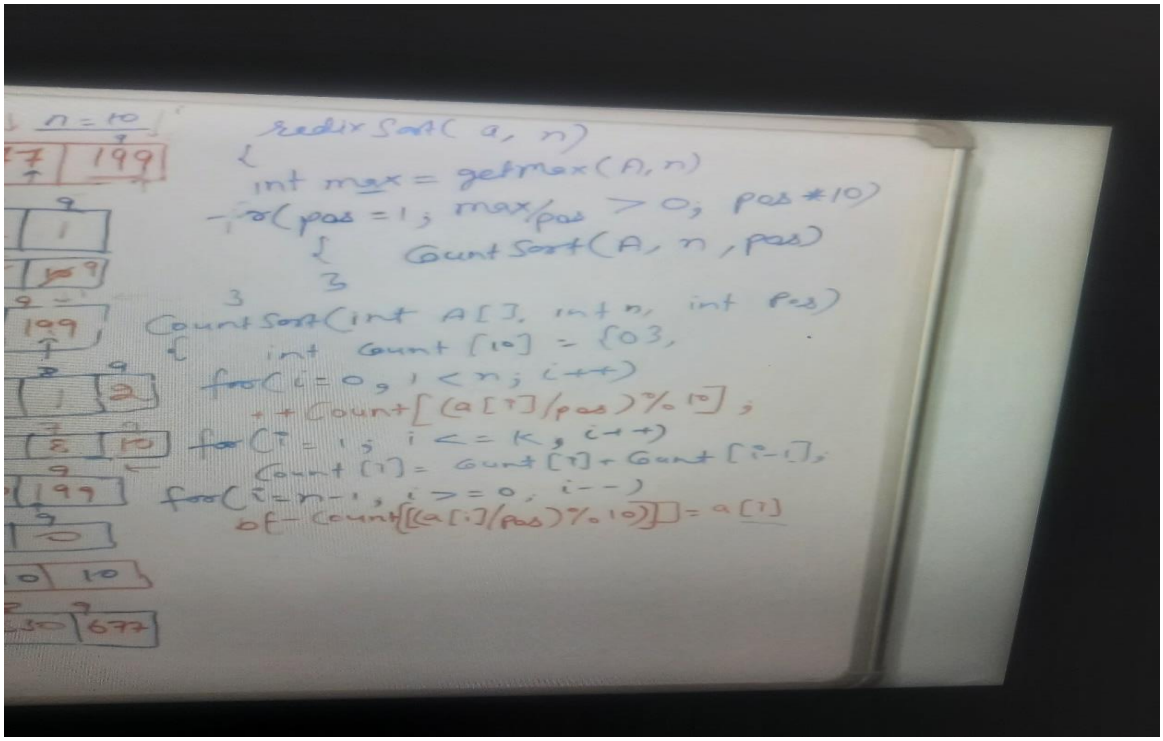


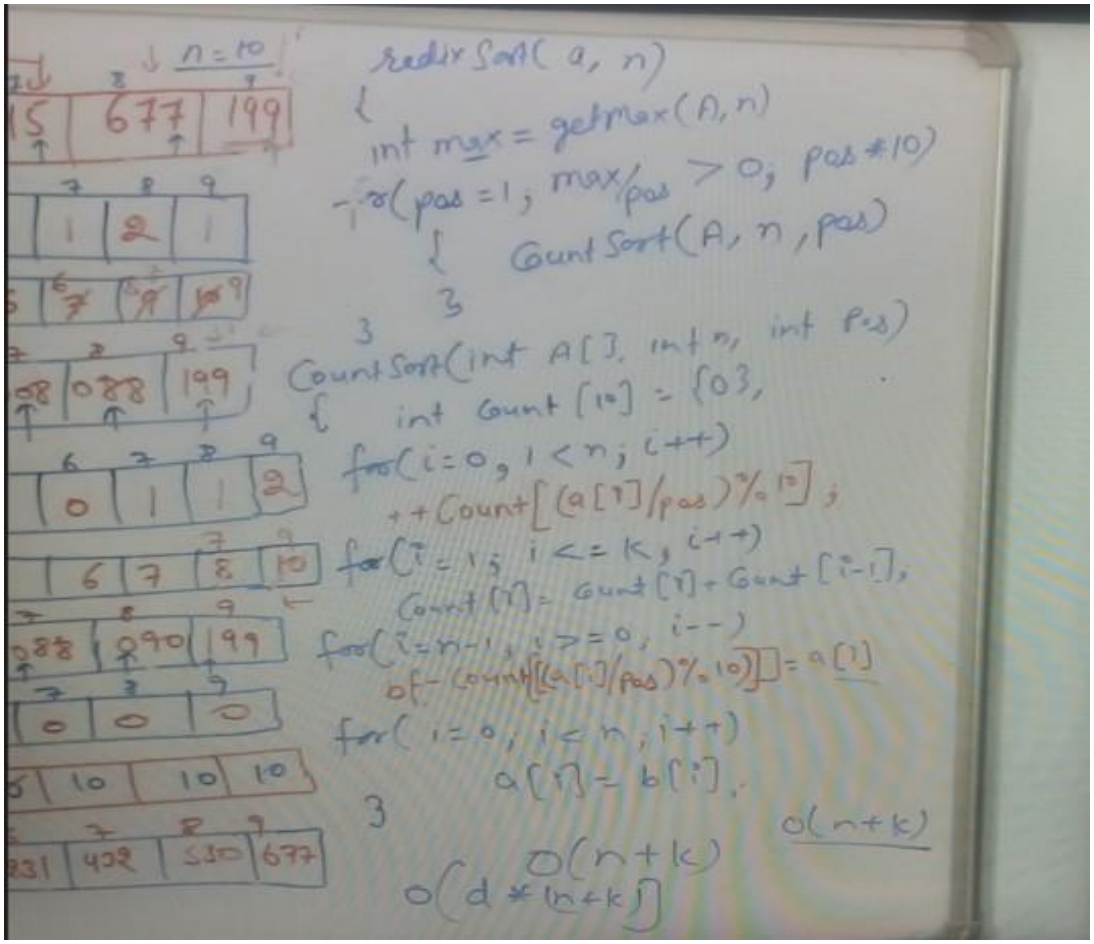
Radix/Bucket sort:

https://www.youtube.com/watch?v=Il45xNUHGp0&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=107

https://github.com/SushantPoman/DataStructures/blob/master/src/RadixSort.java







2D Array:

https://www.youtube.com/watch?v=KDQXUysHLL8&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLl8LU&index=5