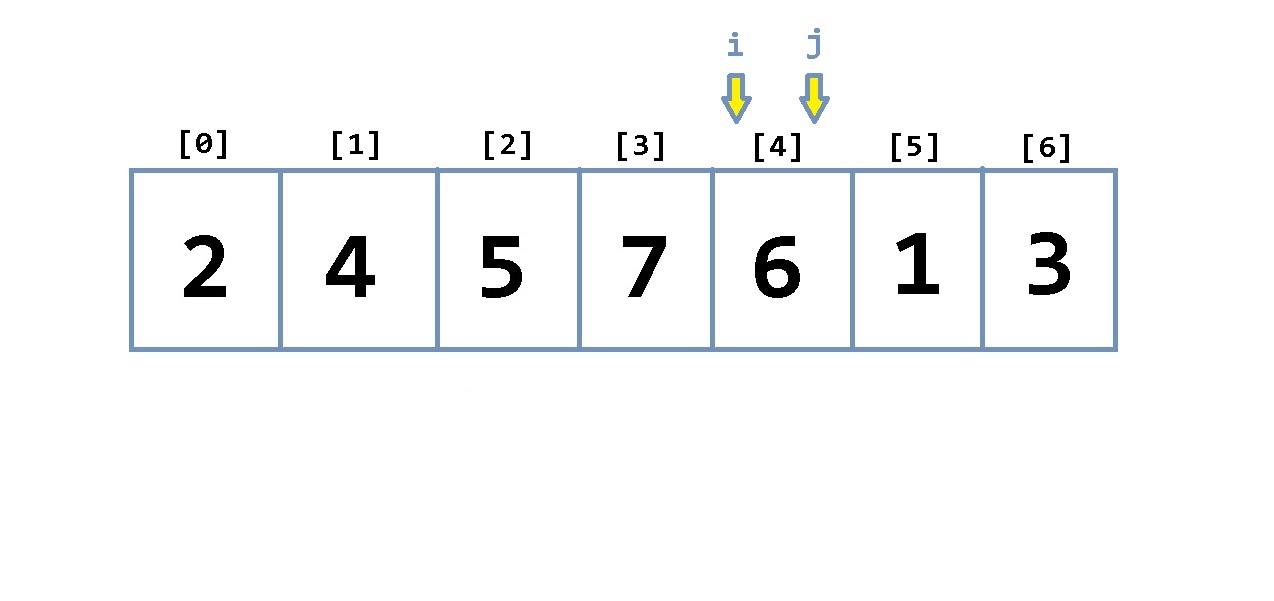
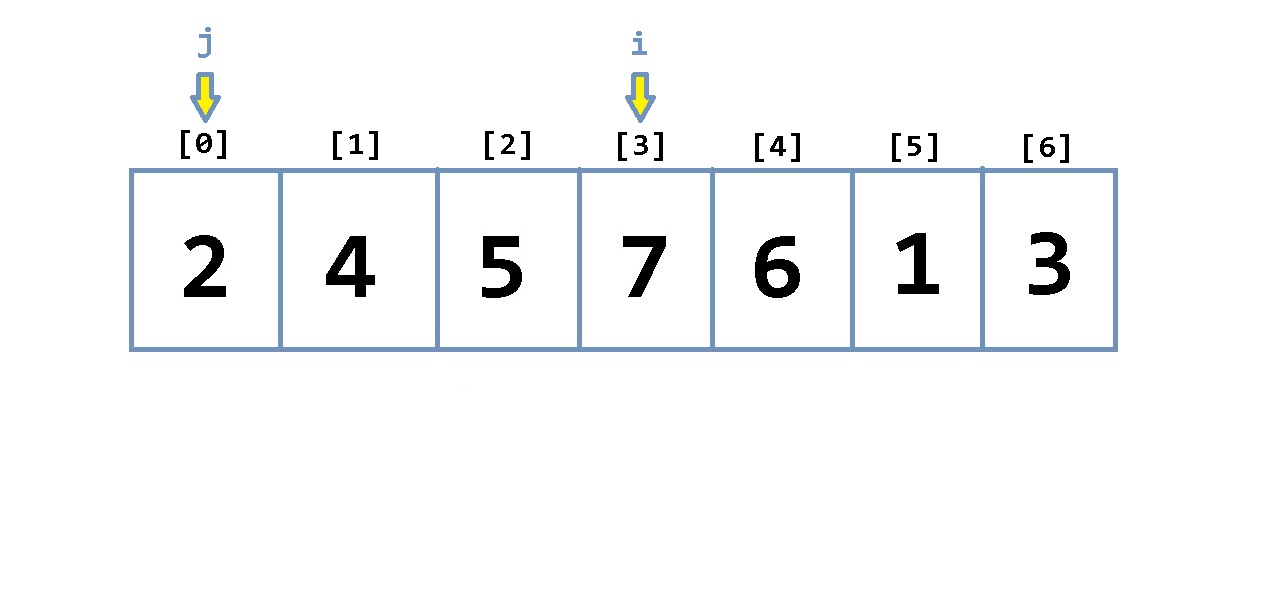
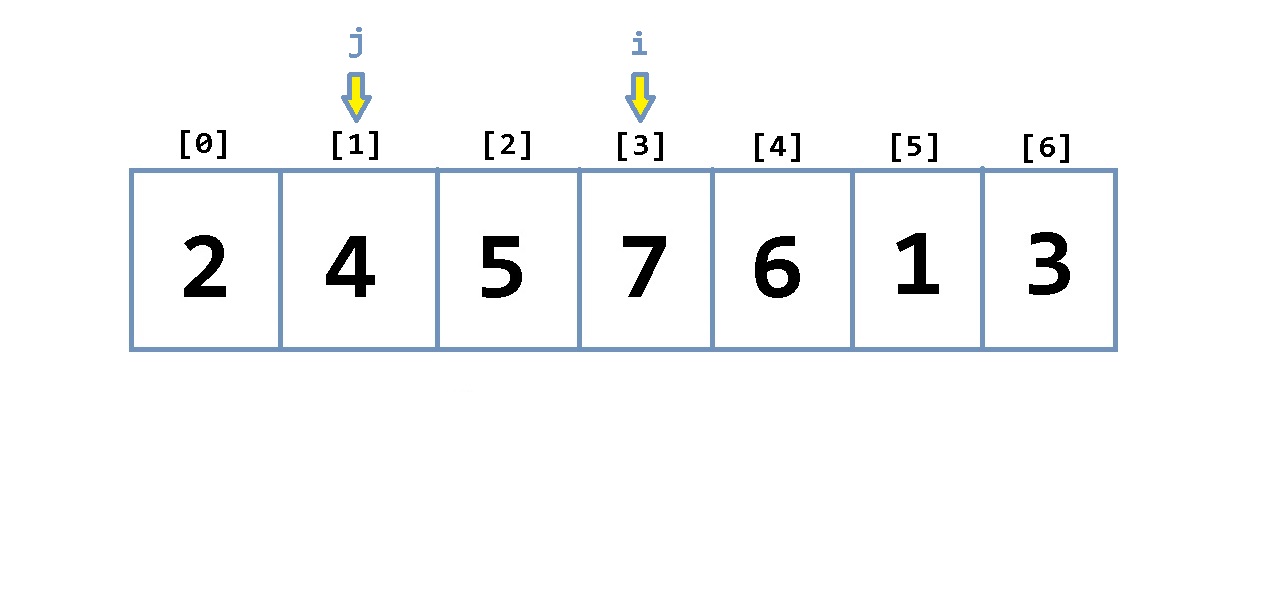
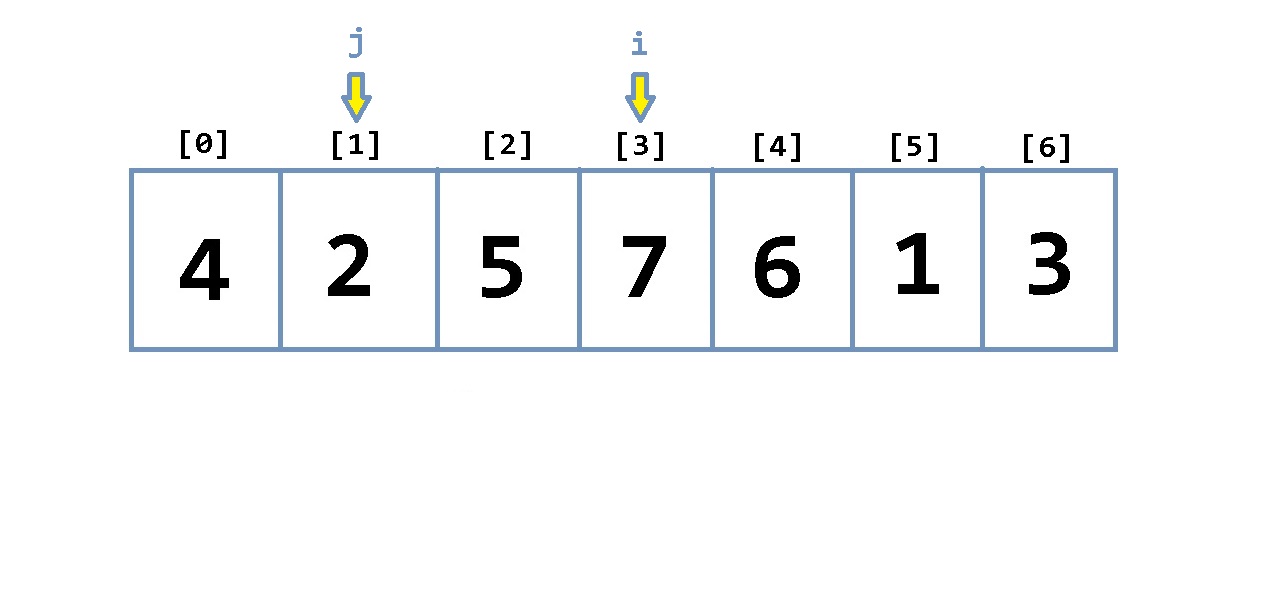
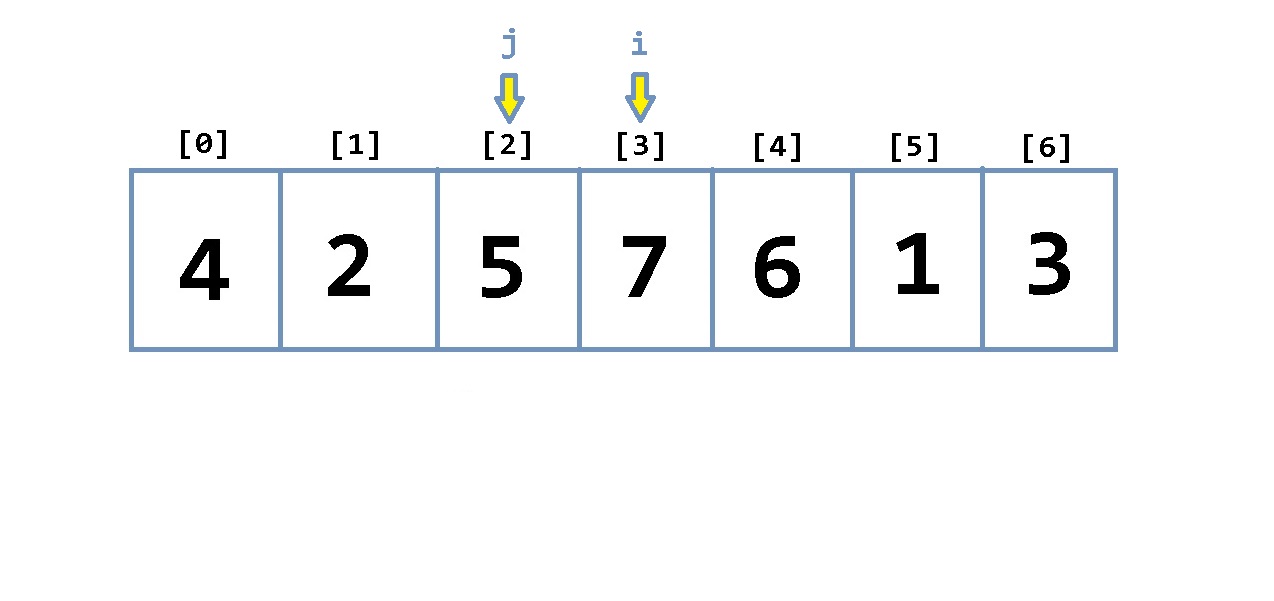
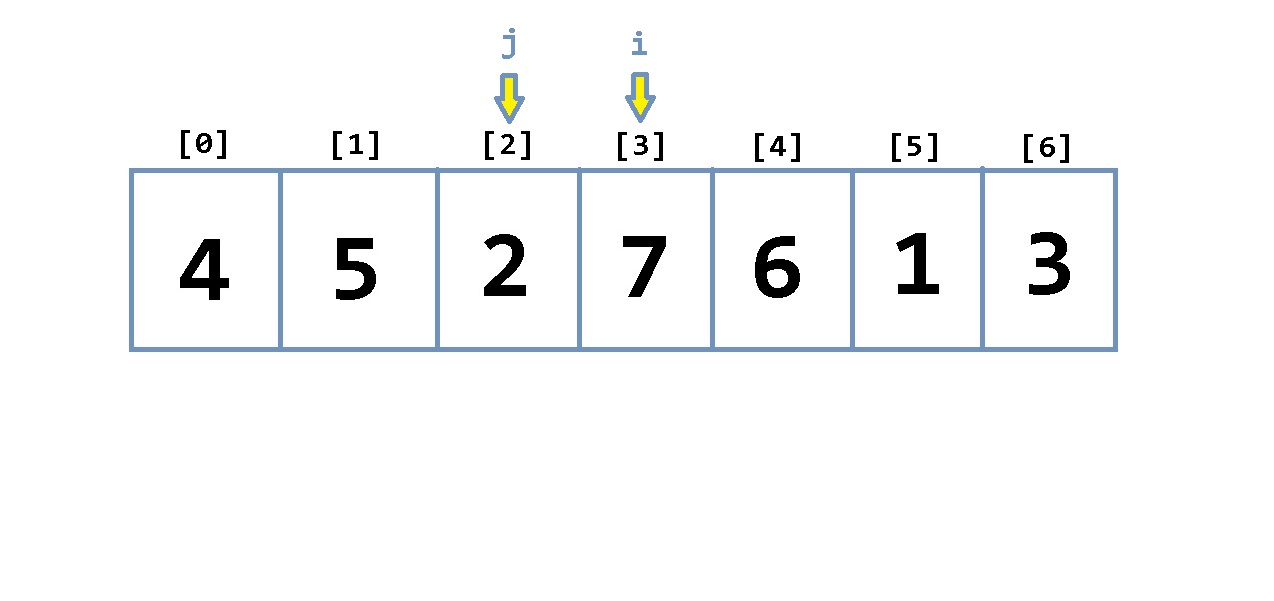
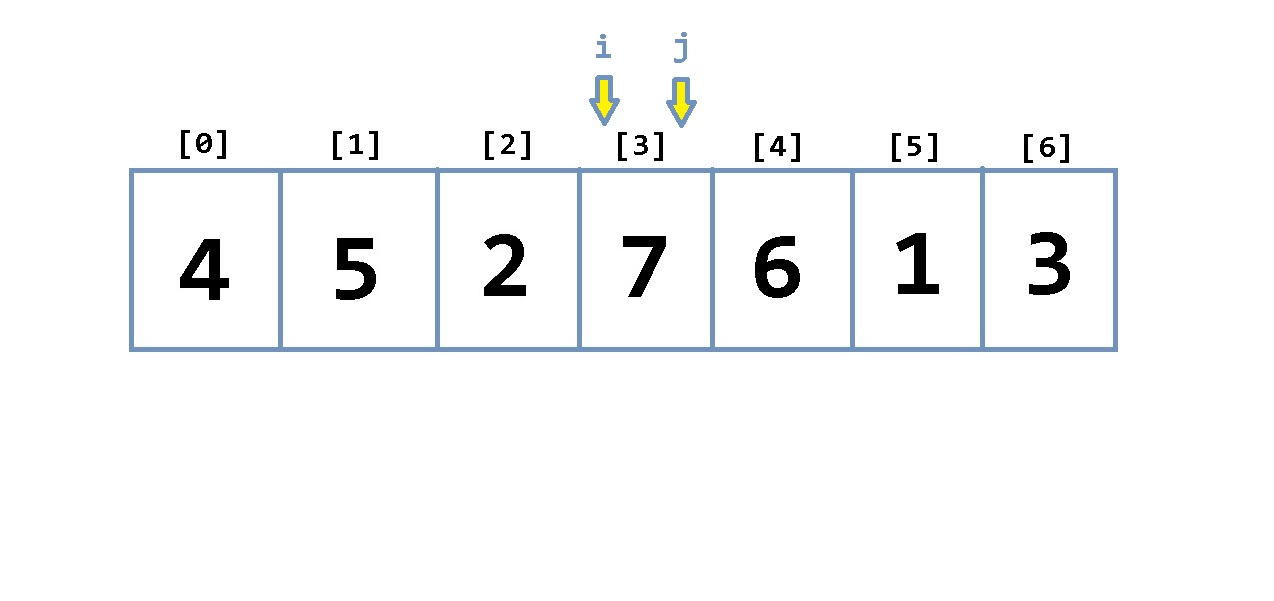
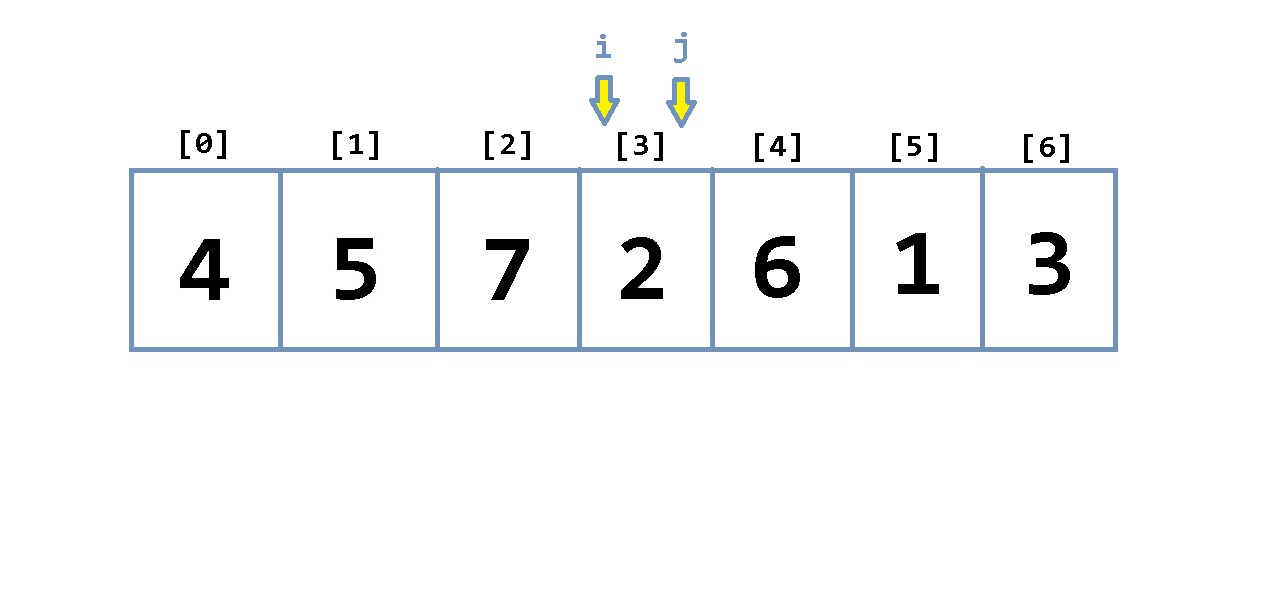
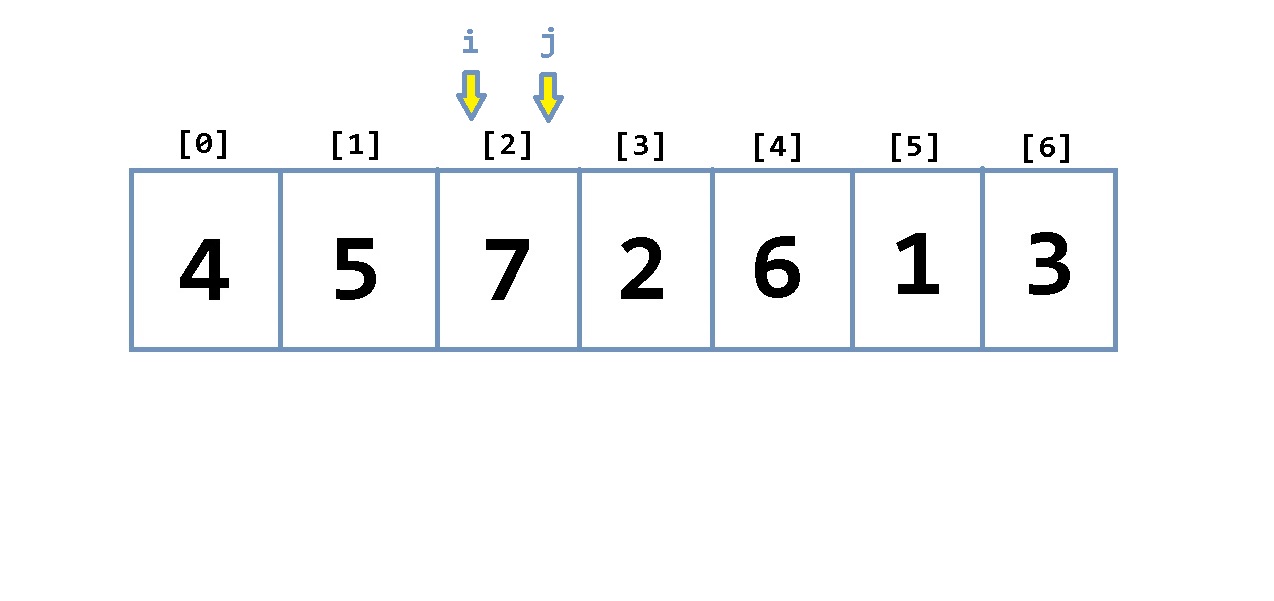
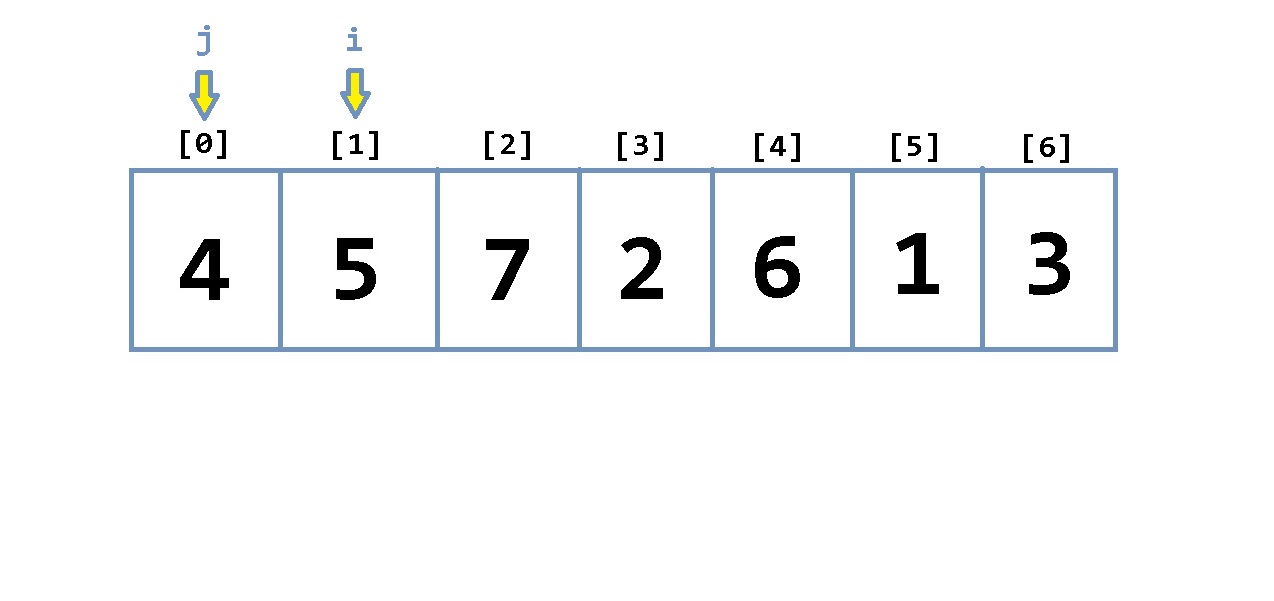
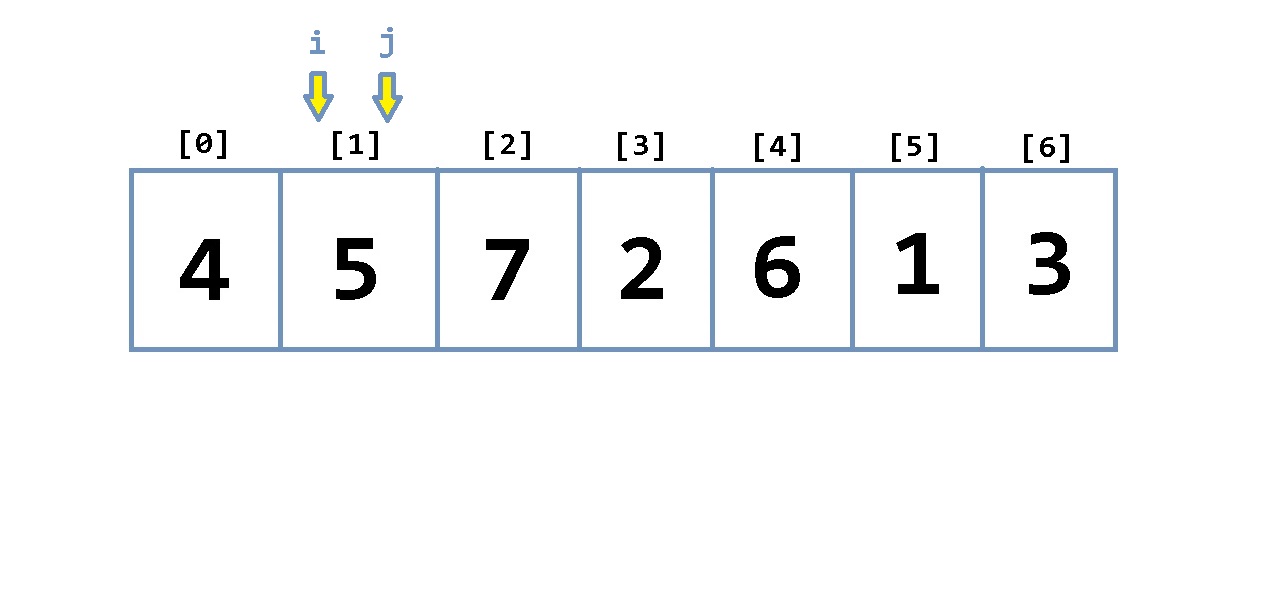
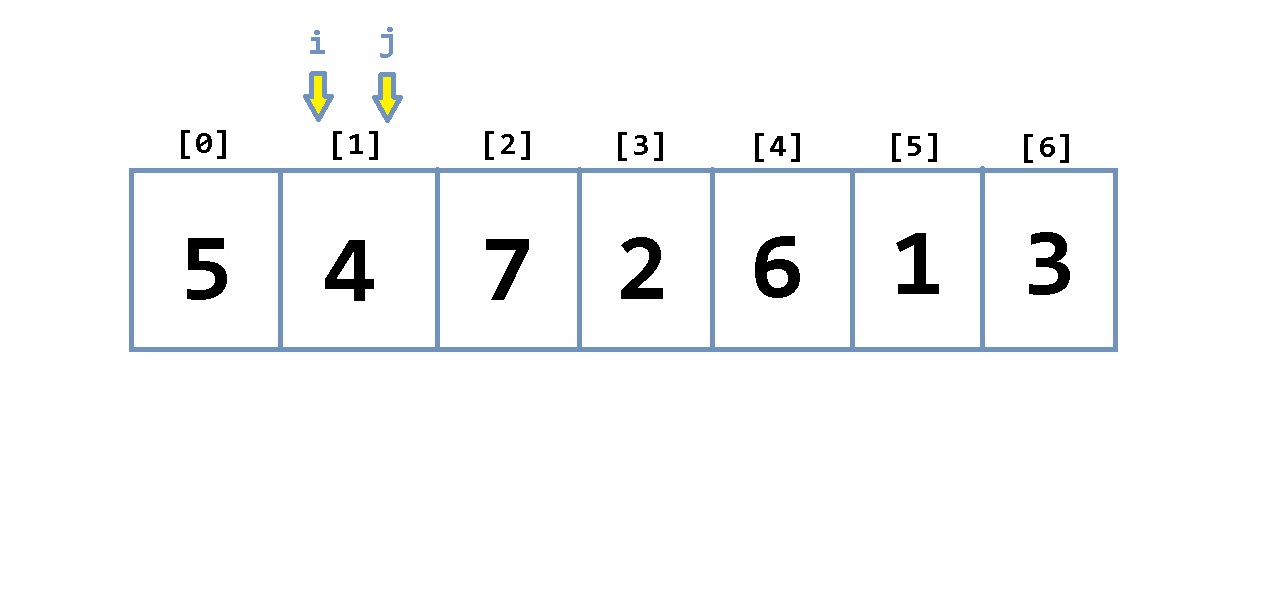
Insertion Sort

Web Resources:

<https://en.wikipedia.org/wiki/Insertion_sort>

<http://cforbeginners.com/insertionsort.html>

Insertion sort simply moves through each index of an array using two reference points, i and j. The point i tells the sort where it is working from and moves toward the end of the array with each iteration. The point j starts at the point i each iteration and then moves backwards to the beginning of the array “pulling” it’s number with it until the number before it is not larger than the one it is on. Each iteration moves forward until it goes through the entire array at which point it is sorted.



(<http://cforbeginners.com/insertionsort.html>)

Screenshot of sorted 10000 element array by insertion sort:



Merge Sort

Web Resources:

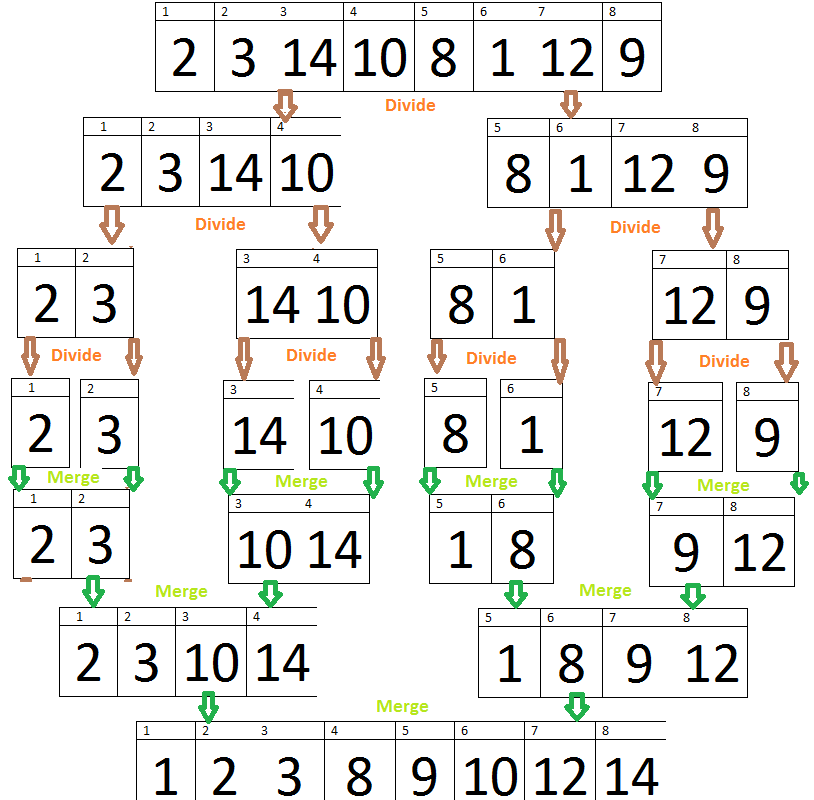
<https://en.wikipedia.org/wiki/Merge_sort>

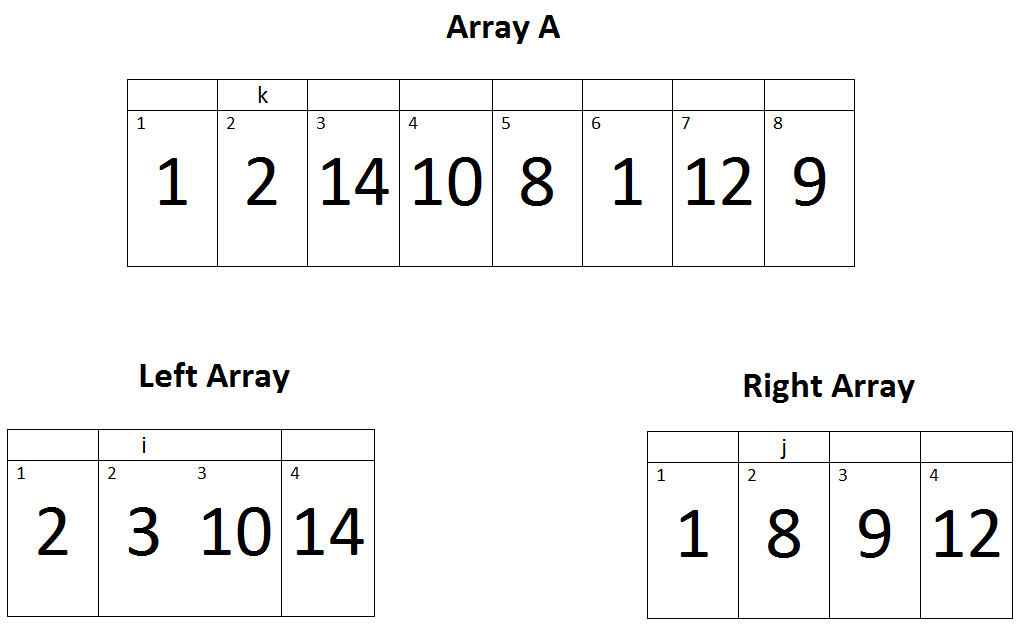
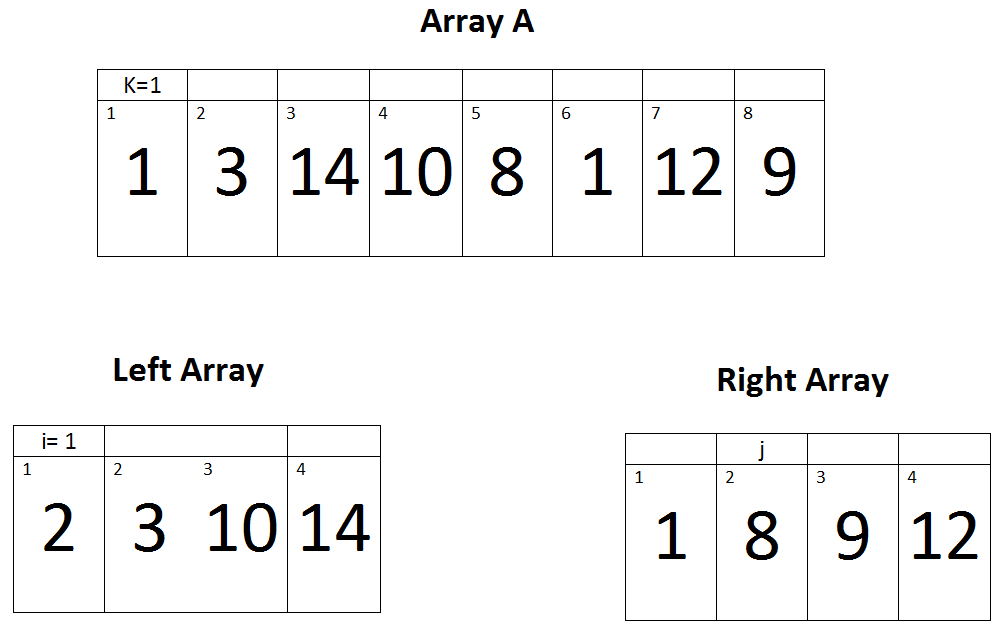
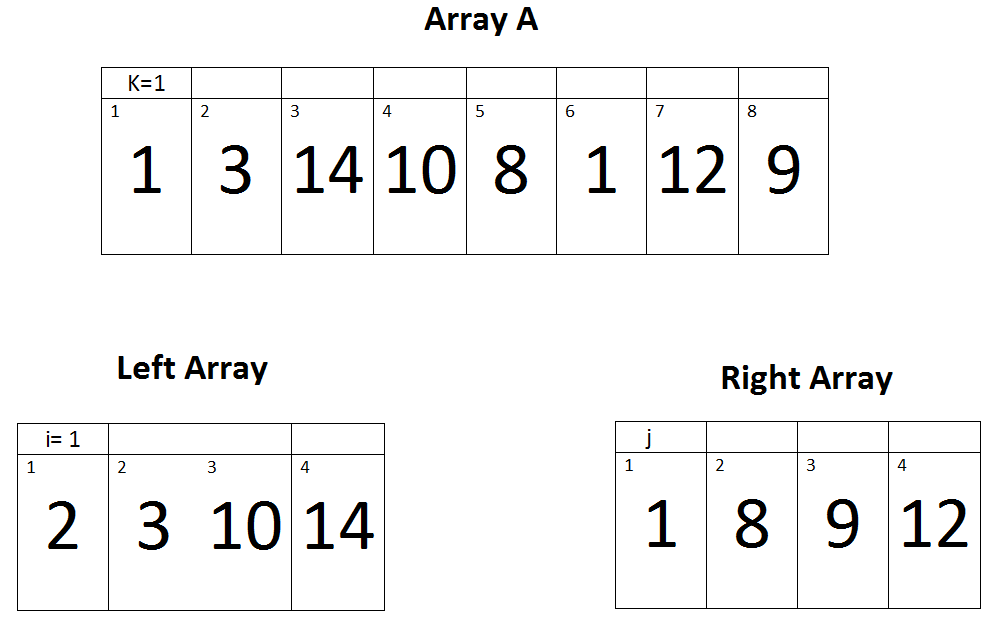
<http://quiz.geeksforgeeks.org/merge-sort/>

<https://www.youtube.com/watch?v=kHYAV3qTcxw>

<http://www.hellgeeks.com/merge-sort/>

Merge sort, compared to linear sort is much more complicated in practice, but the results are much faster for larger data sets. Merge sort essentially has two functions, the merge\_sort function and a merge function. The merge\_sort function is fairly simple in that it checks to make sure the array is not empty and then determines a mid-point of the array. It will then call itself recursively on the first and second halves of the array it has been given, and then call the merge function to put them back together. This recursive calling of itself will happen until the array is completely split and only has one element in each “block” of itself. The merge function essentially checks the two arrays it has been given and orders them before combining them back together. How it does this is that it essentially takes the left and right halves of the array and uses an array reference point, i and j, for each one. It then checks whether i <= j, and depending upon the answer, increments i or j respectively. While it is checking these, it has a reference point, k, on the main array, and it will swap the array based on these two checks. These functions will resolve recursively backwards and at the end the array is sorted.





(<http://www.hellgeeks.com/merge-sort/>)

Screenshot of sorted 10000 element array by merge sort:

