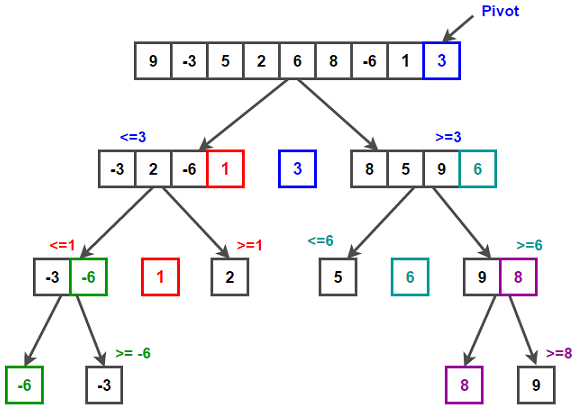
**Quick sort sorting technique**

Quick sort is an sorting technique were divide and conquer rule is been applied. A pivot element is selected in the array and compared wid the current index variable and swapped accordingly with the present index variable. This technique is an inscale were we dont use extra memory for sorting a sorted array indeed sorted array is stored in original array itself.

****

**Working of merge sort:**

1. **Consider the end element of the array as a pivot element.**
2. **We assume 2 indexes one index as a current index and another one has a swapping index.**
3. **Comparison is done between current index and pivot element.**
4. **If our requirement of sorting is from smaller to bigger element then if the current index element is lesser then pivot element then the current index element is swapped with swapping index and both current index and swapping index are incremented by one, or else only current index is incremented.**
5. **At the end of 1st pass the pivot element is in its exact position, and we have 2 set of sub arrays where the lesser elements then pivot element are on left side and greater elements are on right side of pivot element.**
6. **So same procedure is done until we get a sorted list of arrays.**

**Function to quick sort (pseudo code):**

Quicksort(A as array, start as int, end as int){

if (start >= high){

pivot\_location = Partition(A,start,end)

Quicksort(A,start, pivot\_location)

Quicksort(A, pivot\_location + 1, end)

}

}

**Function to partition of array**

Partition(A as array, start as int, end as int){

pivot = A[high];

pindex = start;

for i = start to high-1{

if (A[i] <= pivot) then{

swap(A[i], A[pindex])

pindex = pindex + 1

}

}

swap(pivot,A[pindex])

return (pindex)}