**Quick sort** is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays. A large array is partitioned into two arrays one of which holds values smaller than specified value say pivot based on which the partition is made and another array holds values greater than pivot value.

The quick sort partitions an array and then calls itself recursively twice to sort the resulting two subarrays. This algorithm is quite efficient for large sized data sets as its average and worst case complexity are of O(nlogn) where n is no. of items.

How does it work?

Choose the highest index value has pivot.

The pivot value divides the list in to two parts. And recursively we find pivot for each sub-lists until all lists contains only one element.

JAVA Implementation:

int partition(int arr[], int left, int right)

{

      int i = left, j = right;

      int tmp;

      int pivot = arr[(left + right) / 2];

      while (i <= j) {

            while (arr[i] < pivot)

                  i++;

            while (arr[j] > pivot)

                  j--;

            if (i <= j) {

                  tmp = arr[i];

                  arr[i] = arr[j];

                  arr[j] = tmp;

                  i++;

                  j--;

            }

      };

      return i;

}

void quickSort(int arr[], int left, int right) {

      int index = partition(arr, left, right);

      if (left < index - 1)

            quickSort(arr, left, index - 1);

      if (index < right)

            quickSort(arr, index, right);

}