

QUICK SORT

★ Recursive Sorting Algorithm
 $O(n \log n)$ time complexity

Steps : Pick a Pivot in the array
(can be any element)

Place the pivot in its correct place. Then, place all smaller elements to the pivot's left and all larger elements go to the pivot's right. Repeat this process recursively for left half and right half.

To find the correct position of the pivot within the array, we use two pointer approach. From left try finding first element that is bigger than the pivot, from the right try finding first element smaller than the pivot. After finding both swap them. Repeat this process until the left/right pointers cross each other and that crossing point is where we place our pivot.

Pseudocode :

```
qS(arr, low, high) {  
    if (low < high) {  
        pl = pivotPlace(arr, low, high)  
        qS(arr, low, pl)  
        qS(arr, pl + 1, high)  
    } else return i  
}  
  
pivotPlace(arr, low, high) {  
    pivot = arr[low]  
    i = low  
    j = high  
    while (i < j) {  
        while (arr[i] <= pivot && i <= high) {  
            i++  
        }  
        while (arr[j] >= pivot && j >= low) {  
            j--  
        }  
        if (i < j) {  
            swap(arr[i], arr[j])  
        }  
    }  
    swap(arr[low], arr[j])  
    return j  
}
```