



# Quick Sort

☰ Tags	
🕒 Created time	@November 20, 2022 11:56 AM
🕒 Created time 1	@November 20, 2022 11:56 AM
🕒 Last edited time	@November 26, 2022 7:43 AM
⚙️ Status	In progress
🔗 URL	
🔗 URL 1	

Definition - 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 the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value. ([source](#))

Working - we can divide the working of quick sort in 3 steps

1st select a pivot element(it can be first last mid or random between first to last element)

2nd find its correct position of pivot by iterating all the element keeping all the smaller element to left of pivot and larger on right.

3rd replete the process on array from start to pivot and pivot+1 to n-1

```

int conquer(int vec[],int first,int last){
    int frontPointer = first + 1;
    int lastPointer = last;
    while(frontPointer <= lastPointer){
        while(frontPointer <= last && vec[frontPointer] <= vec[first]){
            frontPointer++;
        }
        while(lastPointer >= first && vec[lastPointer] > vec[first]){
            lastPointer--;
        }
        if(frontPointer <= lastPointer)
            swap(vec[lastPointer--],vec[frontPointer++]);
    }
    swap(vec[first],vec[lastPointer]);
    return lastPointer;
}

void quickSort(int arr[], int low, int high)
{
    // code here
    if(low<high){
        int pivot = conquer(arr,low,high);
        quickSort(arr, low,pivot-1);
        quickSort(arr,pivot+1,high);
    }
}

```

5	6	2	-2	6	10	12	15
---	---	---	----	---	----	----	----