Quick Sort Algorithm:-

arr = {5,2,6,4,1,③}

Pivot & Partition → last Element

① Find Pivot (Sorting is done around the pivot element)

② Partition on the base of pivot.

less =< pivot > greater

Find Pivot (2,①) lesser ← ③ → greater (5,6,④) Find Pivot

1,2 ↓ QS(left) [1,2,3,4,5,6] ↓ QS (Right)

↓

Sorted after the Backtracking

4,5,6

5,⑥

5,6

③ QS(left)

QS(Right)

QuickSort

```
void quickSort(arr,st,end){
    if(st<end){
     Pividx = partition(arr,st,end)
     QS(arr,st,Pividx-1)  → Left
     QS(arr,Pividx+1,end)  → Right
```

Partition

```
int Partition(arr,st,end){
idx=st-1, Pivot=arr[end]
  for(j=st; j<end; j++){
   if(arr[j]<=Pivot){
           idx++
   swap(arr[j], arr[idx])
               }
```

idx = else current position

idx = st-1

j = st to end-1

pivot = arr[end]

```
}
idx ++
swap(arr[end], arr[idx])
    return idx;
    }
```

T.C = $O(n \log n)$ → Average/Practical

worstcase = T.C = $O(n^2)$

↓ pivot = smallest, largest

S.C = $O(1)$