QuickSort

Merge sort	Quicksort
↪ Combine	↪ Partition
⇓	↓
Merge Procedure	Divide the array into two subarrays
	↪ No combine required in quickSort

$$mid = l + (h - l)/2$$

$QS(0, 1)$ — $l, m-1$

$n = 6$ — m

$QS(m+1, h)$

0	1	2	3	4	5
~~40~~ 10	20 ~~70~~	~~10~~ 40 ~~60~~	70 ~~20~~	60 ~~10~~	90
~~i~~	J	J	J	J	J
x	~~i~~	i			

$l = 0$

$h = 5$

$x = \text{Pivot} = arr[l] = 40$

$i \rightarrow$ to take smaller elements than pivot

$J \rightarrow$ to take larger element than pivot

QS(3, 5)

3	4	5
~~70~~ 60	~~60~~ 70	90
x	J	J
	i	

```
Partition (arr, l, h) {
    i = l
    pivot = arr(l);
    for(j = l+1; j <= h; j++) {
        if (arr(j) <= pivot) {
            i = i+1;
            swap(arr(i), arr(j));
        }
    }
    Swap(arr(l), arr(i))
    return i;
```

O(n)

correct index of pivot element → Swap(arr(l), arr(i))

Note

1. Pivot Element reaches to its correct position
2. left side of pivot — smaller element

   Right side of pivot — bigger element

QuickSort Algorithm

function name $\rightarrow$ quickSort (arr, l, h) {

if (l < h) {

1. Divide the array

n ——— m = partition(arr, l, h);

2. Conquer the subproblems via recursion

$m-1-l+1$

$T(m-l)$ ——— quickSort(arr, l, m-1);

↳ Recursion

$T(h-m)$ ——— quickSort(arr, m+1, h);

}

$h-(m+1)+1$

$h-m-\cancel{1}+\cancel{1}$

Recurrence Relation

$$T(n) = \begin{cases} 1 & ; n=1 \\ \underline{T(\boxed{m-l})} + \underline{T(h-m)} + n & ; n>1 \end{cases}$$

↳ Partition

① QS(0,2) — Left subarray | QS(4,6) — Right subarray | algorithm

0	1	2	3	4	5	6
10	30	40	50	70	60	90

QS(0,0) QS(2,5)

0	1	2	3	4	5
10	50	70	60	65	90

Best case & Average case

$$T(n) = T(n/2) + T(n/2) + n$$

$$= 2T(n/2) + n$$

$$= O(n \log n)$$

Worst case

$$T(n) = T(n-1) + n$$

$$= O(n^2)$$

Insertion sort → $O(n)$

Sorted / Almost sorted

Nothing

0	1	2	3	4
10	20	30	40	50
x, i	J	J	J	J

Quicksort → Geniune scenario

↳ Highly unsorted

Space complexity

↓

Inplace sorting

↓

$O(1)$

↳ Not a stable algorithm

$$\boxed{\begin{array}{ccccc} 10' , & 10'' , & 10''' , & 10'''' , & 10''''' \\ 0 & 1 & 2 & 3 & 4 \end{array}}$$

↳ Randomized QuickSort

↳ Pivot element

(Randomly chosen)

0	1	2	3	4	5
~~7~~ 5	4	6	~~5~~ 7	1	3