# Quick Sort

$\{6, 3, 9, 5, 2, \overset{\text{Pivot}}{\boxed{8}}\}$ - Pivot & partition

↓

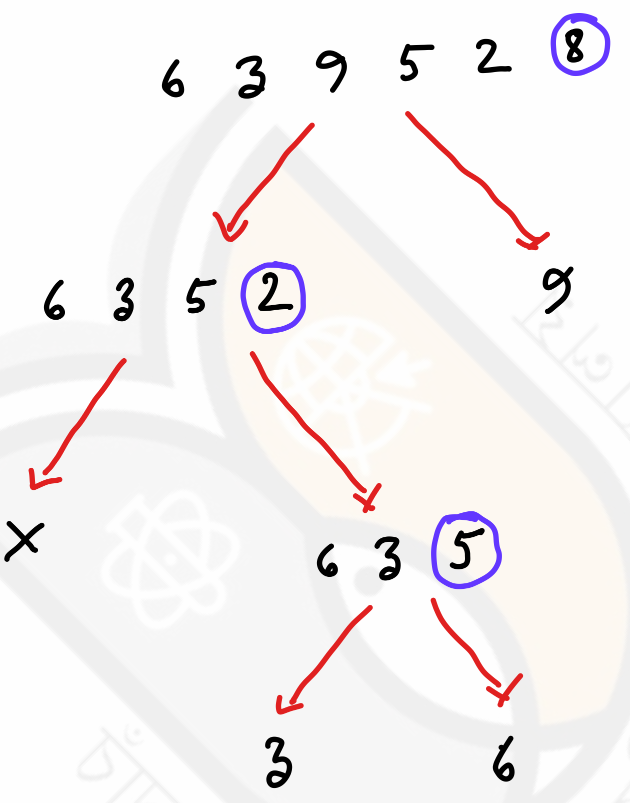
Central point

↓

we can take pivot as

- Random
- Median
- 1st element
- last element

○ = Pivot



left → Pivot → Right

2 3 5 6 8 9

## Code:

low=0 high=5

6 3 9 5 2 ⑧ → Pivot

i = -1 } I will create space for the value which is smaller then pivot Initially we are assuming -1 became of There is no such element.

6 3 9 5 2 ⑧ → Pivot

6 is smaller then Pivot

so i becomes 0 and create place for 6

6

i=0

3 is smaller then Pivot
so i becomes 1 and create place for 3

6 3
i=0 i=1

9 is greater then Pivot
so i remains 1 and will not create a place

6 3 9
i=0 i=1

5 is smaller then Pivot
so i becomes 2 and create place for 5 and Swap the Place with 9

6 3 5 9
i=0 i=1 i=2

2 is smaller then Pivot
so i becomes 3 and create place for 2

6 3 5 2 9
i=0 i=1 i=2 i=3

* As we placed all the smallest element compared to pivot
so if we increment i pivot will find his actual place

6 3 5 2 8 9
i=0 i 1 i-2 i=3 i=4

8

9

| 6 3 5 2 |
|---|

again Quick sort

## Time Complexity

Worst case → $O(n^2)$
Average → $O(n \log n)$

* Worst case occurs when pivot is always the smallest or largest element

* we take the last element as pivot so if our array arranged in Ascending or descending order Worst Case will be happened

* we didn't use any extra array in Quick sort like merge sort

1, 2, 3, 4, 5 → n-1
1,2,3,4 → n-2
1,2,3 → n-3

$n + n-1 + n-2 \ldots$

$\frac{n(n+1)}{2}$

$O(n^2)$