

Quick sort:

```
QuickSort(arr[], l, r) {
```

```
    if (l < r)
```

```
    {
```

```
        int p = Partition(arr[], l, r);
```

```
        QuickSort(arr, l, p-1); // Left
```

```
        QuickSort(arr, p+1, r); // Right
```

```
    }
```

arr[]

l

r

p

pivot = 7

Example:

6	3	9	5	2	8	7
6	3	5	2	7	8	9
2	6	3	5	7	8	9
2	3	5	6	7	8	9
2	3	5	6	7	8	9

1. low অবসরস্থ করা হবে না।

2. i দিয়ে সের করা হবে
Position যেখানে pivot
element থাকে।

3. j দিয়ে i এর সঠিক-
value খুঁজে সের করা
হবে।

```
Partition (int array[], int low, int high) {
```

```
    int Pivot = array[high]
```

```
    int i = (low - 1);
```

```
    for (int j = low; j < high; j++)
```

```
    {
```

```
        if (array[j] <= Pivot)
```

```
        {
```

```
            i++;
```

```
            swap (&array[i], &array[j]);
```

```
        }
```

swap(array[i+1], array[high]);

// pivot-নিজের জায়গা নেয়।

} ~~return (i+1);~~

} return (i+1)

4. এঁ দিয়ে সবগুলো নম্বর চেক করি।
i collect করে রাখে কোথায়
pivot নম্বর আছে।

এঁ যখন চেক করে তখন swap
করে swap(array[i], array[i]);
যাতে pivot থেকে ছোট নম্বর গুলো
pivot এর বামদিকে জায়গা পায়।

Pivot = 7

New array,

6 7

6 3 *

6 3 5 *

6 3 5 2 7