

1. Pick the Pivot Element.

1. Place the pivot element.
2. Place the pivot in the correct position of the sorted array.

4 3. smaller on the left & larger on the right.

Pseudo code:

```
int Partition (vector<int>&arr, low, high)
```

$\text{pivot} \leftarrow \text{arr}[\text{low}]$

i ← low

j ← high

```
while (i < j)
```

```
while (arr[i] <= pivot && i < high)
```

 $i++;$

while (arr[j] >= pivot && j > low)

j - i

if $(i < j)$

swap(arr[i], arr[j])

swap (pivot, arr[j])

```
return j;
```

.

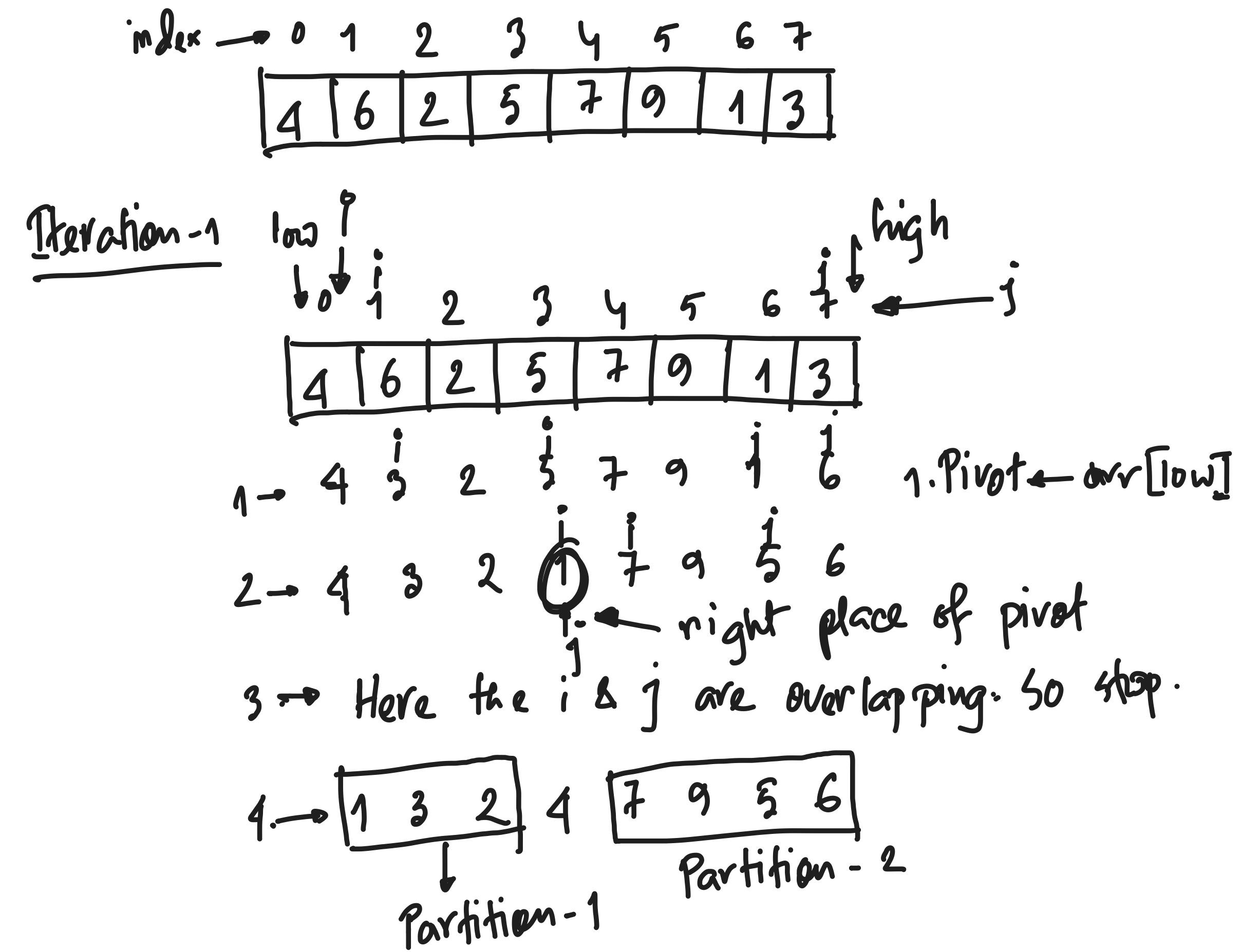
```
void QuickSort(vector<int>& arr, low, high)
```

if (low < high)

$\text{pIndex} \leftarrow \text{Partition}(\text{arr}, \text{low}, \text{high})$

Quick Sort (arr, low, p_index - 1)

QuickSort(arr, pIndex + 1, high)



index \rightarrow 0 1 2 3 4 5 6 7

4	6	2	5	7	9	1	3
---	---	---	---	---	---	---	---

Iteration-1

low \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow high
0 1 2 3 4 5 6 7 $\leftarrow j$

4	6	2	5	7	9	1	3
---	---	---	---	---	---	---	---

1 → 4 3 2 5 7 9 1 6 1. Pivot ← arr[low]

2 → 4 3 2 1 7 9 5 6
 ← right place of pivot

3 \rightarrow Here the i & j are overlapping. So stop.

4. \rightarrow

1	3	2
---	---	---

 4

7	9	5	6
---	---	---	---

↓
Partition - 1 Partition - 2