QUICK SORT

By Prince Agarwal

[“ Hello World ”]

QUICK SORT

Quick Sort is based on divide and conquer approach

Approach :-

Select one Element randomly called as Pivot element

Then Move it to with the First elements

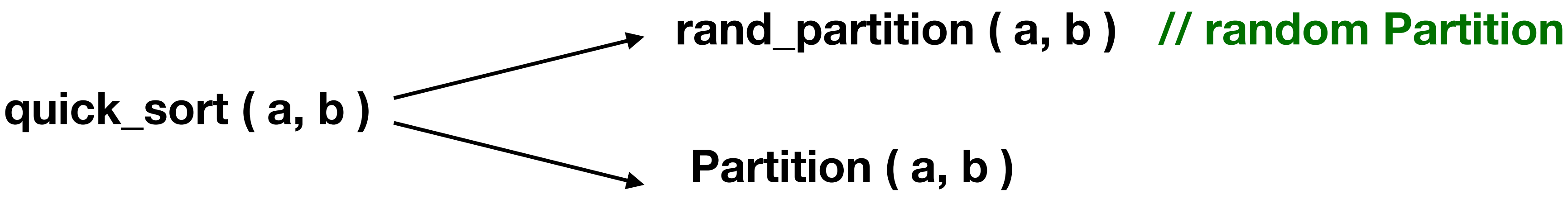
Now ,

Partition the array around the pivot element

Such that, elements which is smaller than the Pivot element
Arrange them in left side

And

Such that, elements which is larger than the Pivot element
Arrange them in right side



rand_partition (a, b) // random Partition

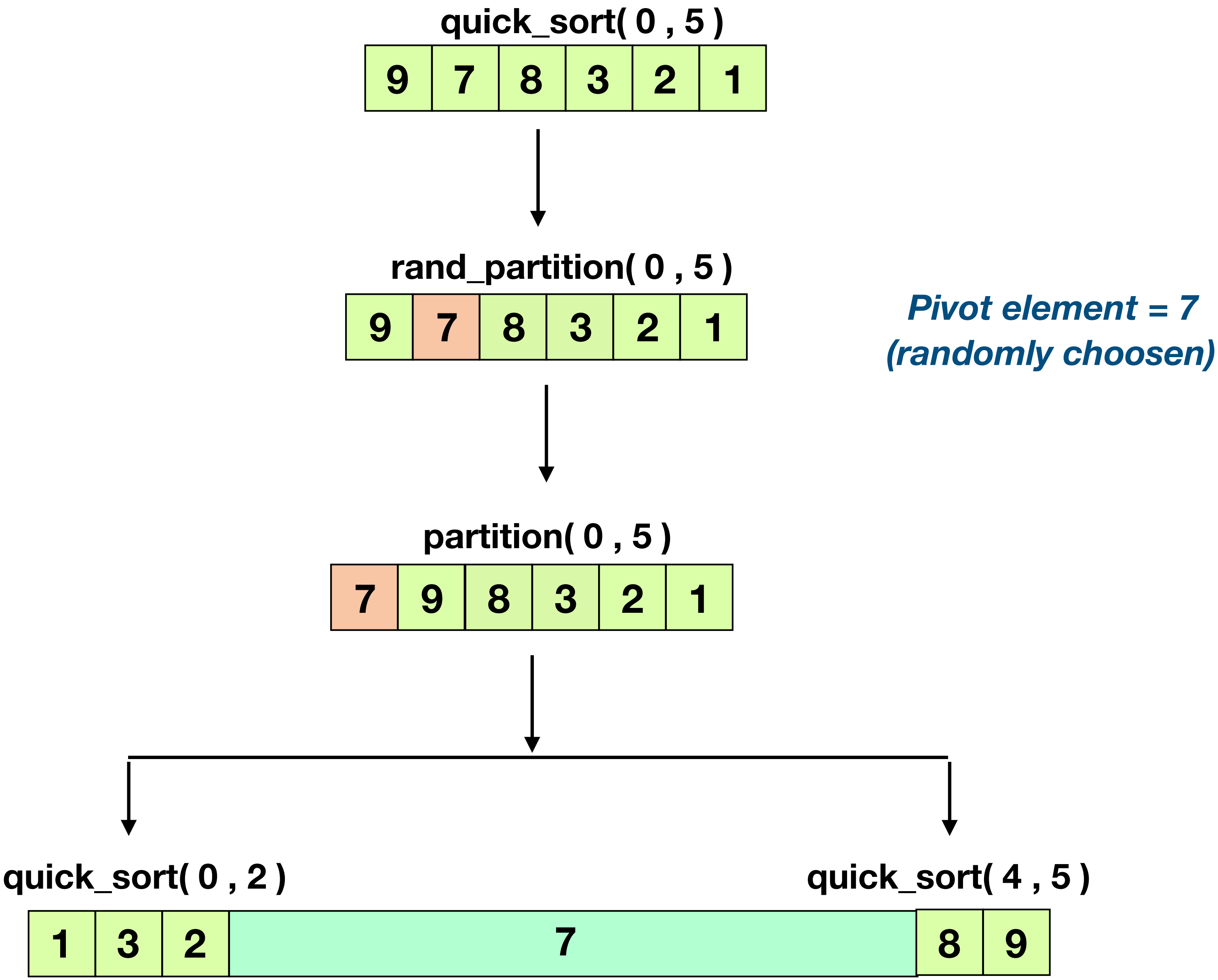
quick_sort (a, b)

Partition (a, b)

QUICK SORT

Sort the Array using QUICK sort :-

9	7	8	3	2	1



quick_sort(0 , 5)

9	7	8	3	2	1

rand_partition(0 , 5)

9	7	8	3	2	1

Pivot element = 7
(randomly choosen)

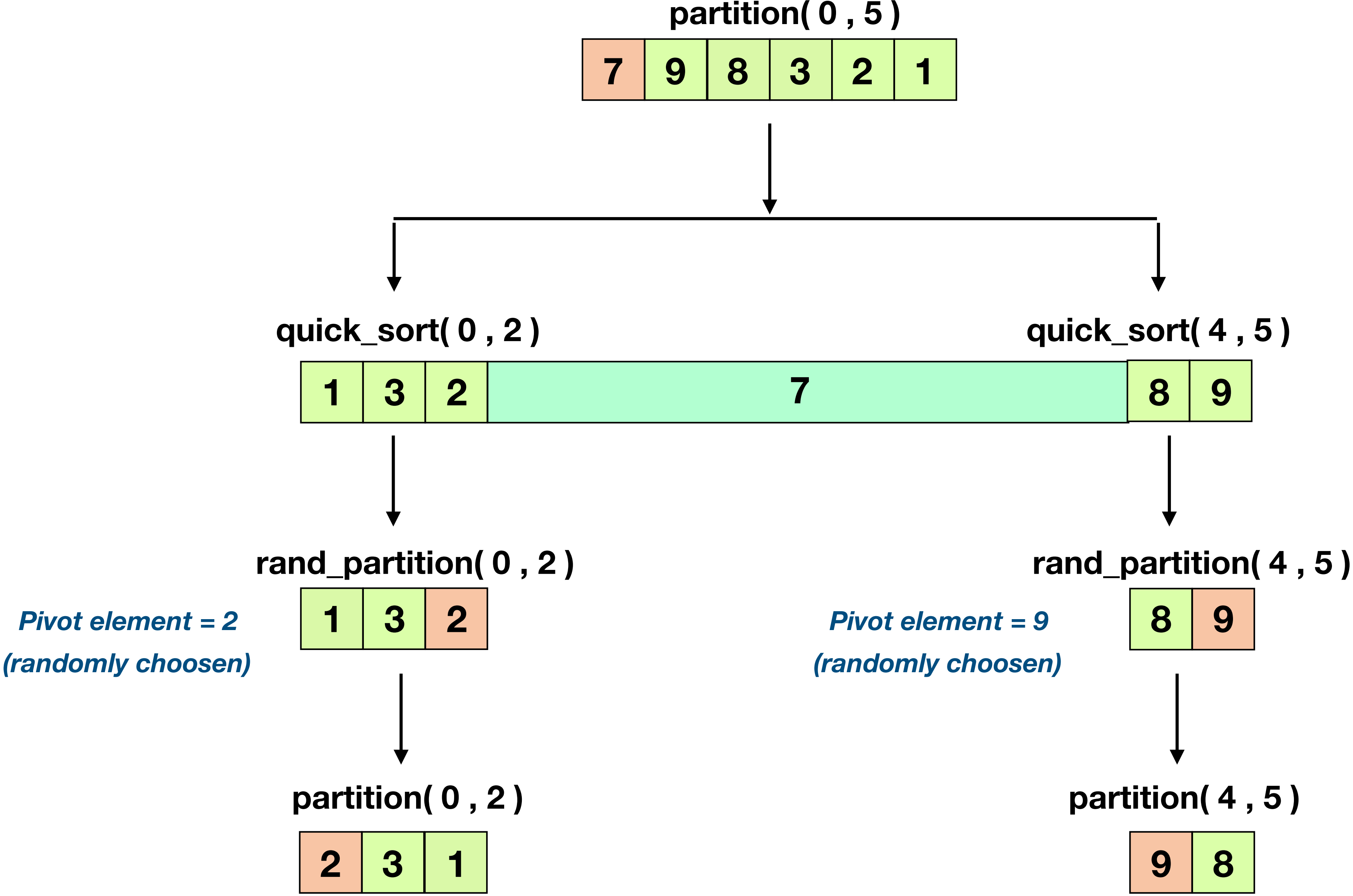
partition(0 , 5)

7	9	8	3	2	1

quick_sort(0 , 2)

quick_sort(4 , 5)

1	3	2	7	8	9



partition(0 , 5)

7	9	8	3	2	1

quick_sort(0 , 2)

quick_sort(4 , 5)

1	3	2	7	8	9

rand_partition(0 , 2)

Pivot element = 2

(randomly choosen)

1	3	2

partition(0 , 2)

2	3	1

rand_partition(4 , 5)

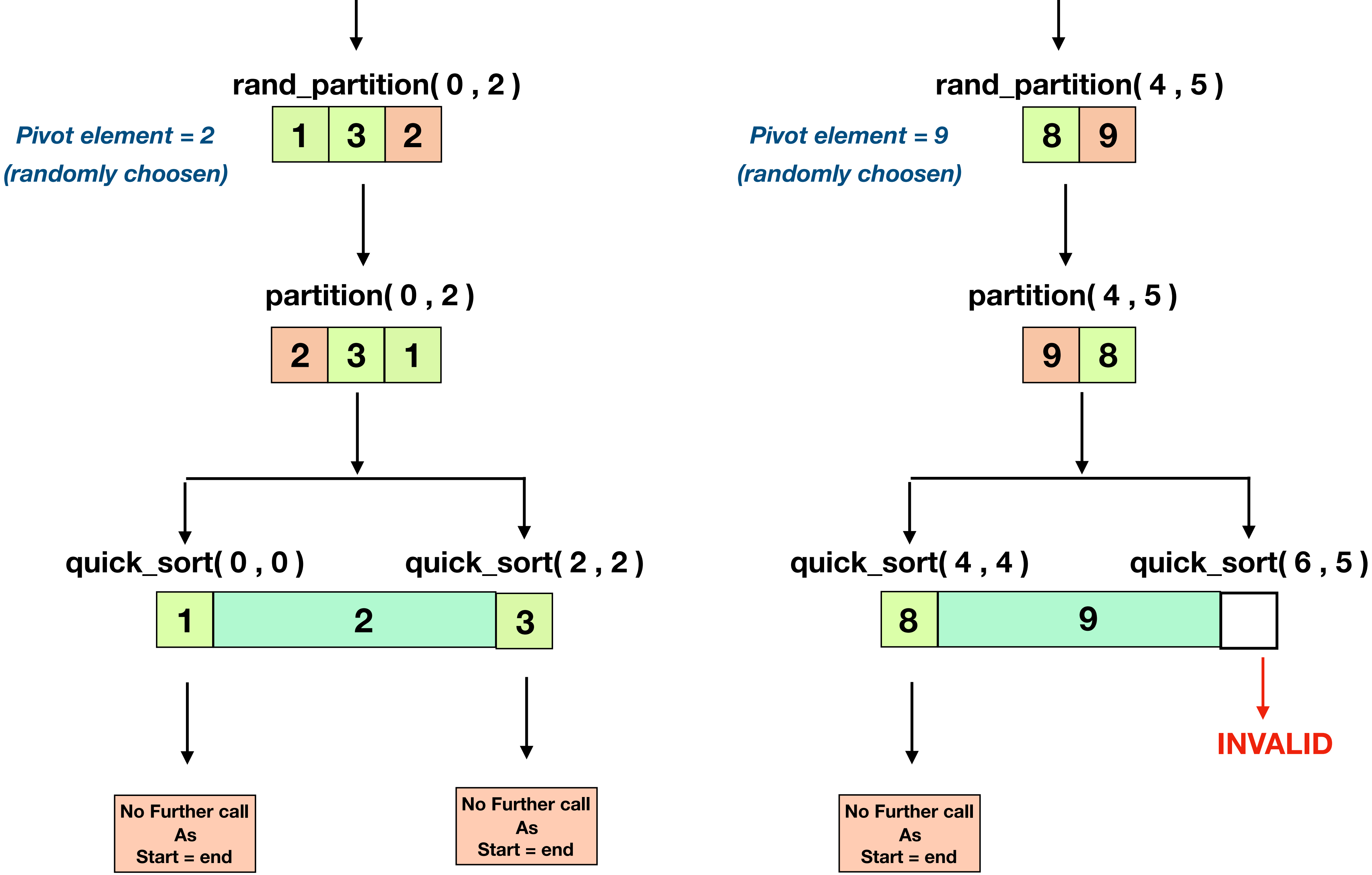
Pivot element = 9

(randomly choosen)

8	9

partition(4 , 5)

9	8



rand_partition(0 , 2)

1	3	2

Pivot element = 2

(randomly choosen)

partition(0 , 2)

2	3	1

quick_sort(0 , 0)

quick_sort(2 , 2)

1	2	3

No Further call
As
Start = end

No Further call
As
Start = end

rand_partition(4 , 5)

8	9

Pivot element = 9

(randomly choosen)

partition(4 , 5)

9	8

quick_sort(4 , 4)

quick_sort(6 , 5)

8	9	

No Further call
As
Start = end

INVALID

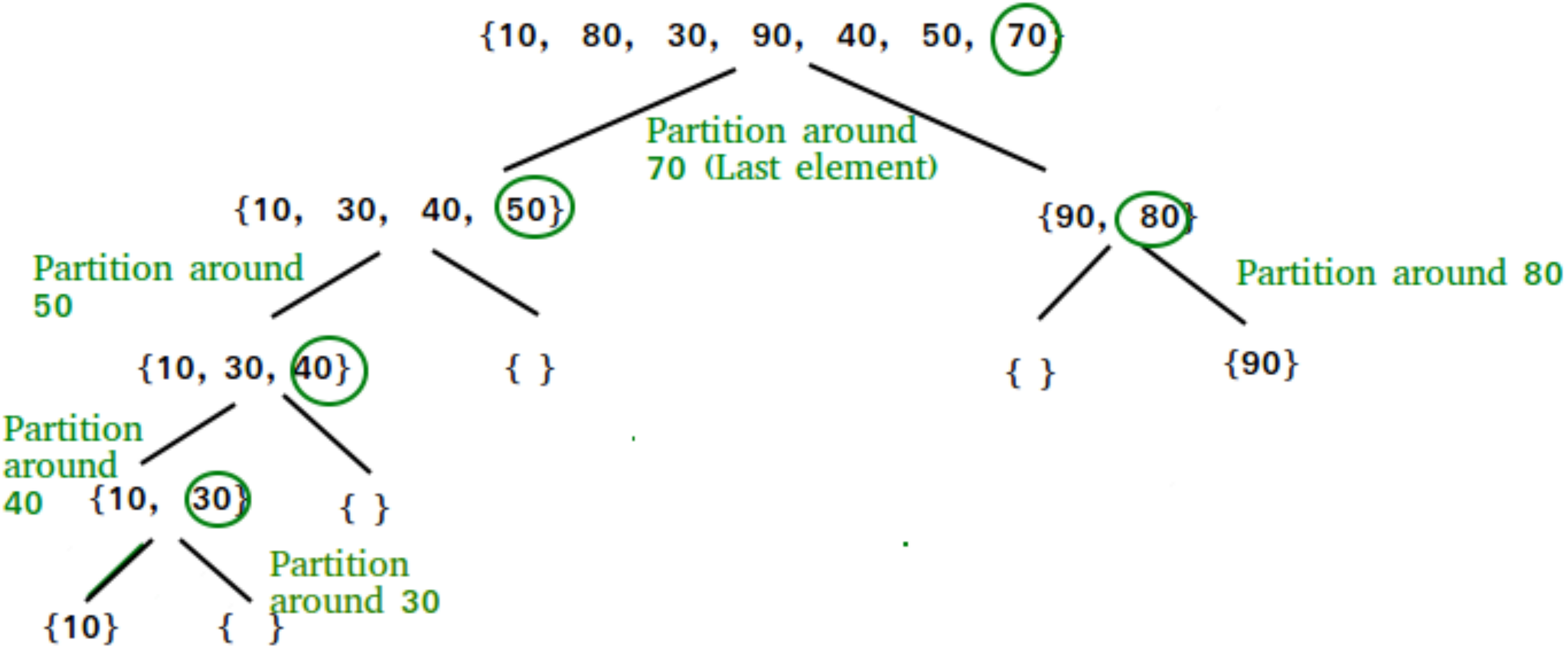
Now, Sorted Array Will be :-

1	2	3	7	8	9

QUICK SORT

Sort the Array using QUICK sort :-

10	80	30	90	40	50	70



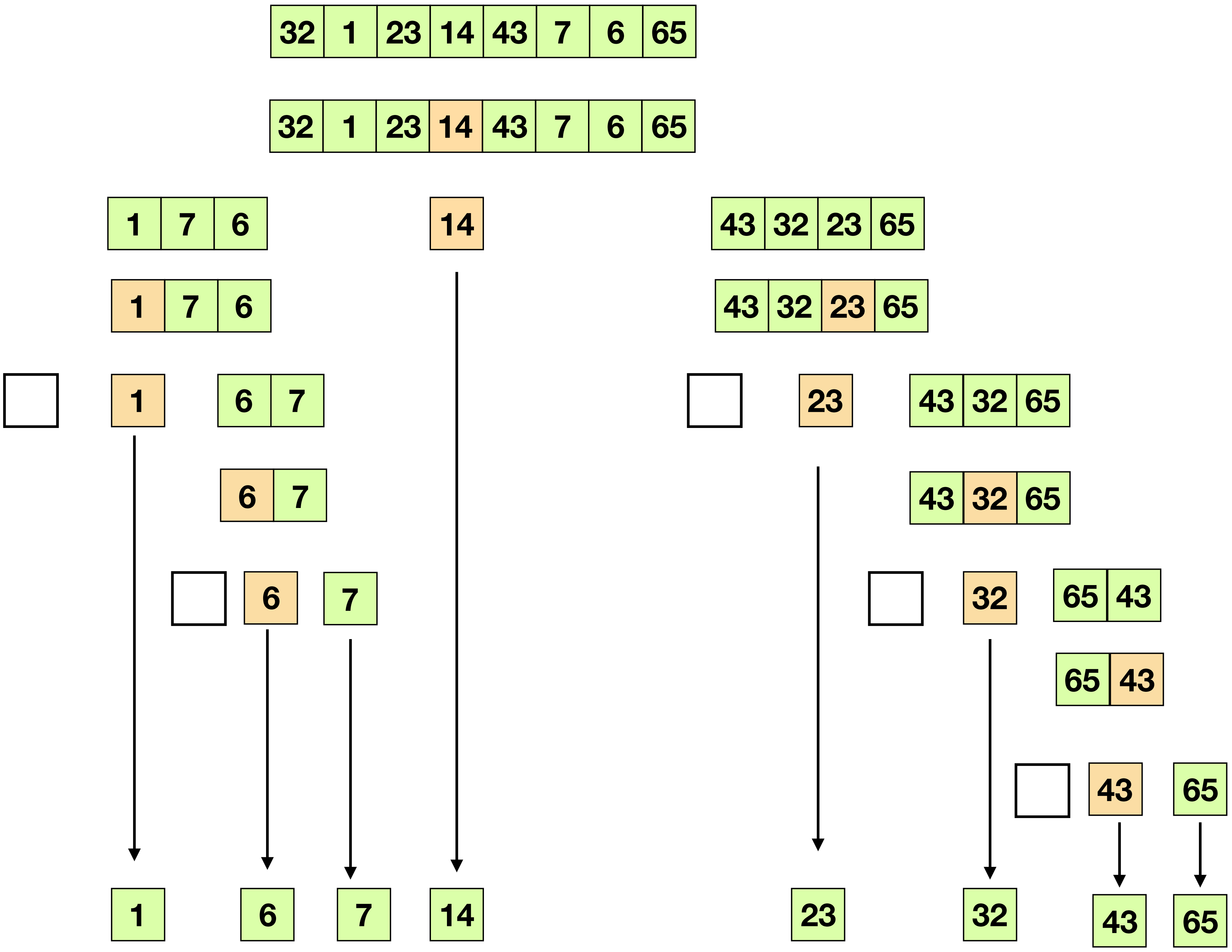
Sorted Array will be : -

10	30	40	50	70	80	90

QUICK SORT

Sort the Array using QUICK sort :-

32	1	23	14	43	7	6	65


32 1 23 14 43 7 6 65
32 1 23 14 43 7 6 65
1 7 6
14
43 32 23 65
1 7 6
43 32 23 65
1
6 7
23
43 32 65
6 7
43 32 65
6
7
32
65 43
65 43
43
65
1
6
7
14
23
32
43
65

QUICK SORT

Time Complexity

Time Complexity = O(N^2) *[Worst time complexity]*

But, This Algorithm is based on our random selection of elements

So the time complexity fluctuates between

$$O(N^2) \text{ and } \underline{O(NlogN)}$$

But in most case it comes out to be →

Subscribe, Like & Share

Hello World

*“ If you feel any problem then comments in my video
I will reply as soon as possible “*

- Prince Agarwal