Randomized Quicksort Example

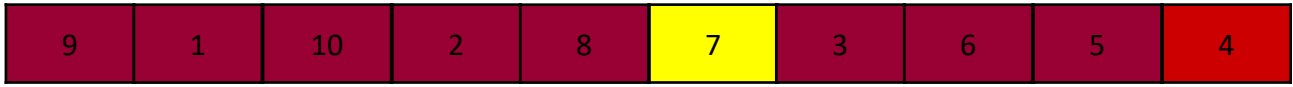
This deck illustrates an example of randomized quicksort

Each step chooses a random pivot in current subarray
swaps it with last item in the subarray
& then partitions

Yellow items will be current pivot.
Grey items will be previous pivots
(currently in correct location)

1st Call

Choose Random Pivot (7)



Swap with last element (4)

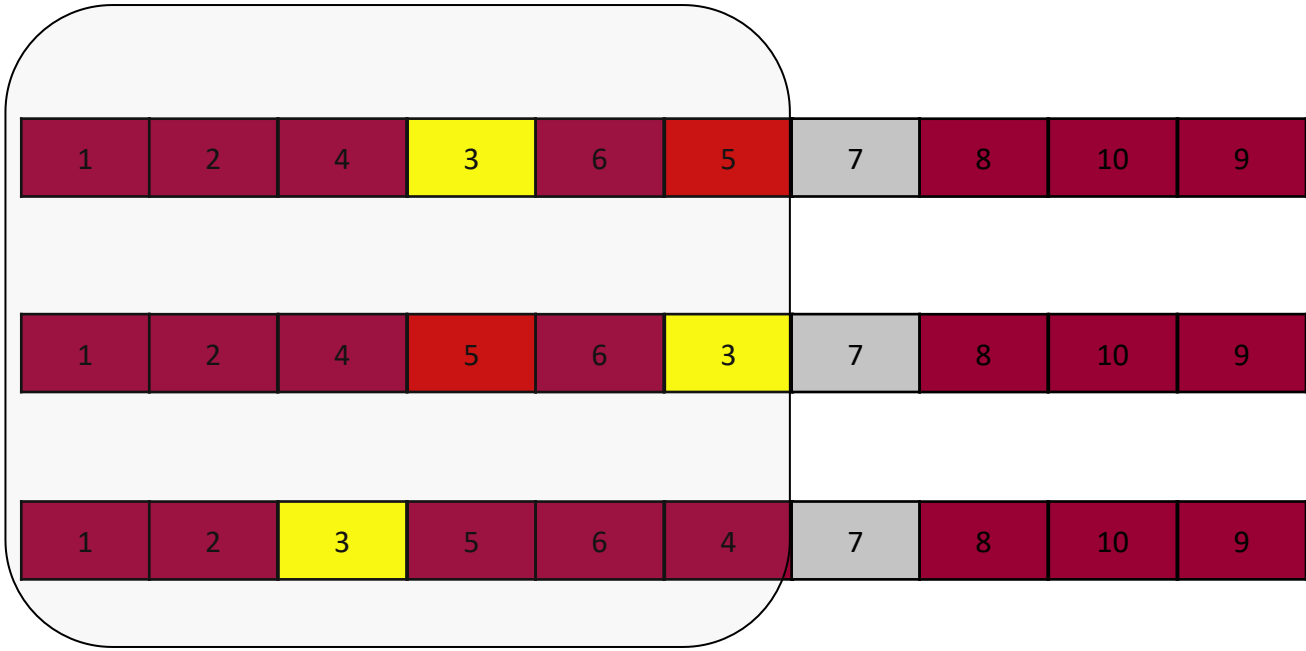
Partition around pivot

2nd Call

Choose Random Pivot (3)

Swap with last element (5)

Partition around pivot

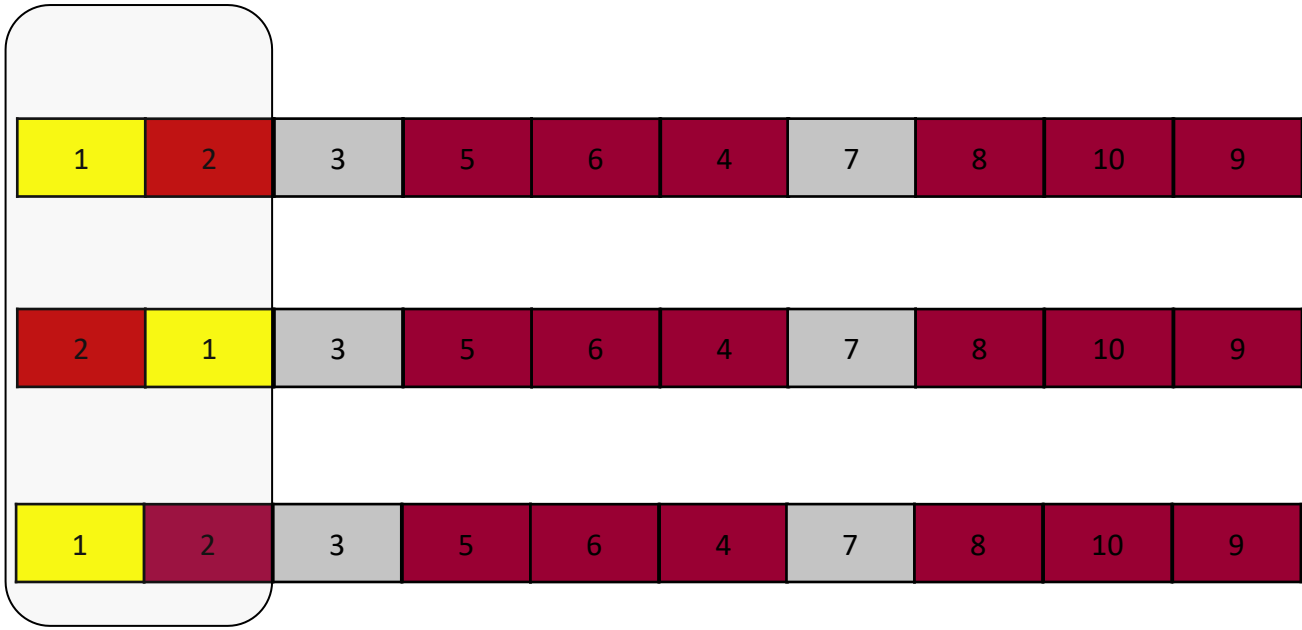


3rd Call

Choose Random Pivot (1)

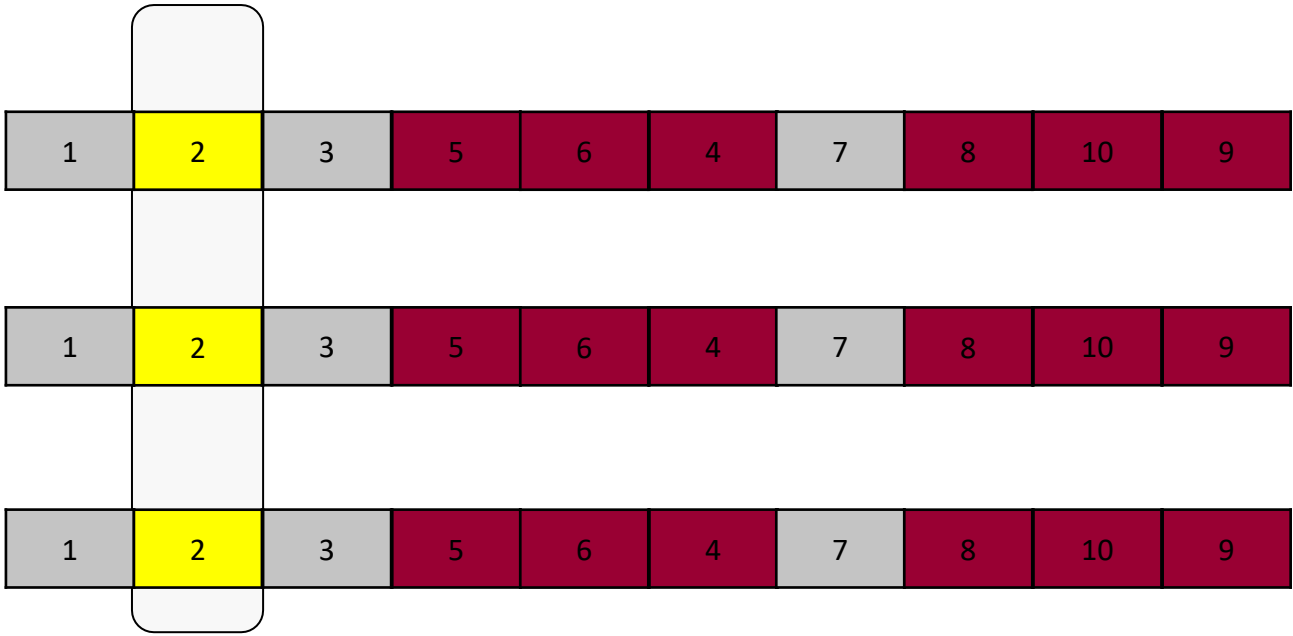
Swap with last element (2)

Partition around pivot



4th Call

Choose Random Pivot (2)



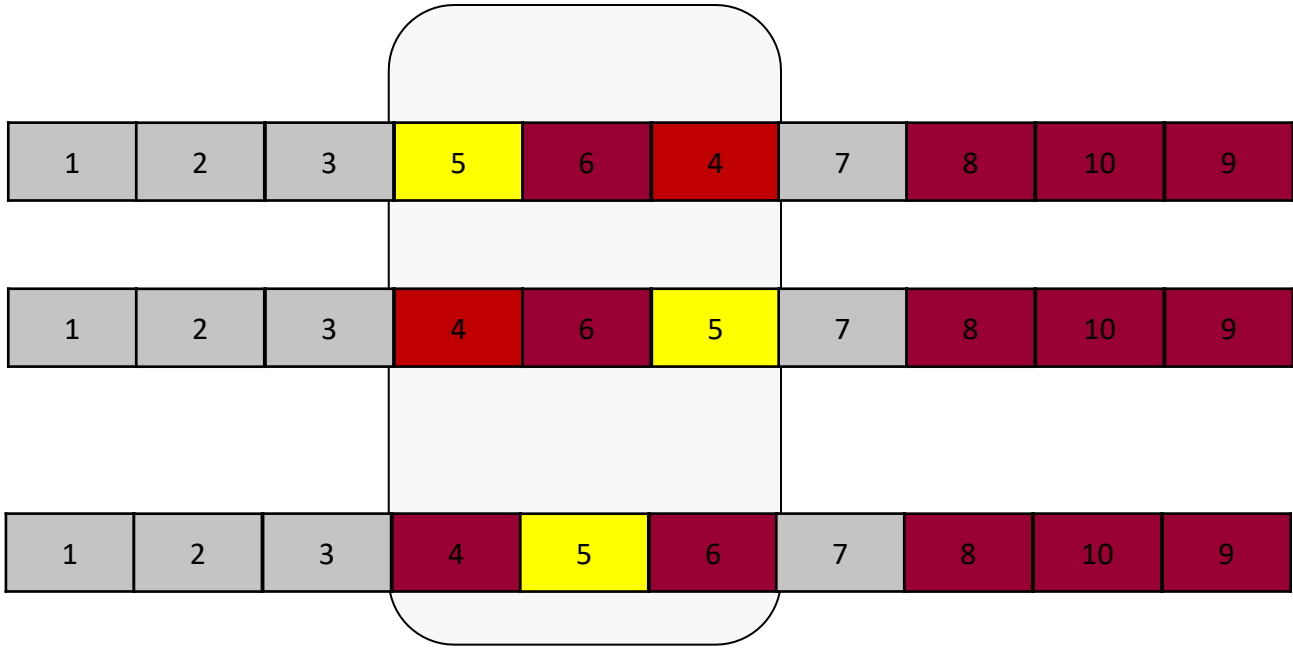
Swap with last element (2)

Partition around pivot

Since subarray contains only one element no work is actually done

5th Call

Choose Random Pivot (5)

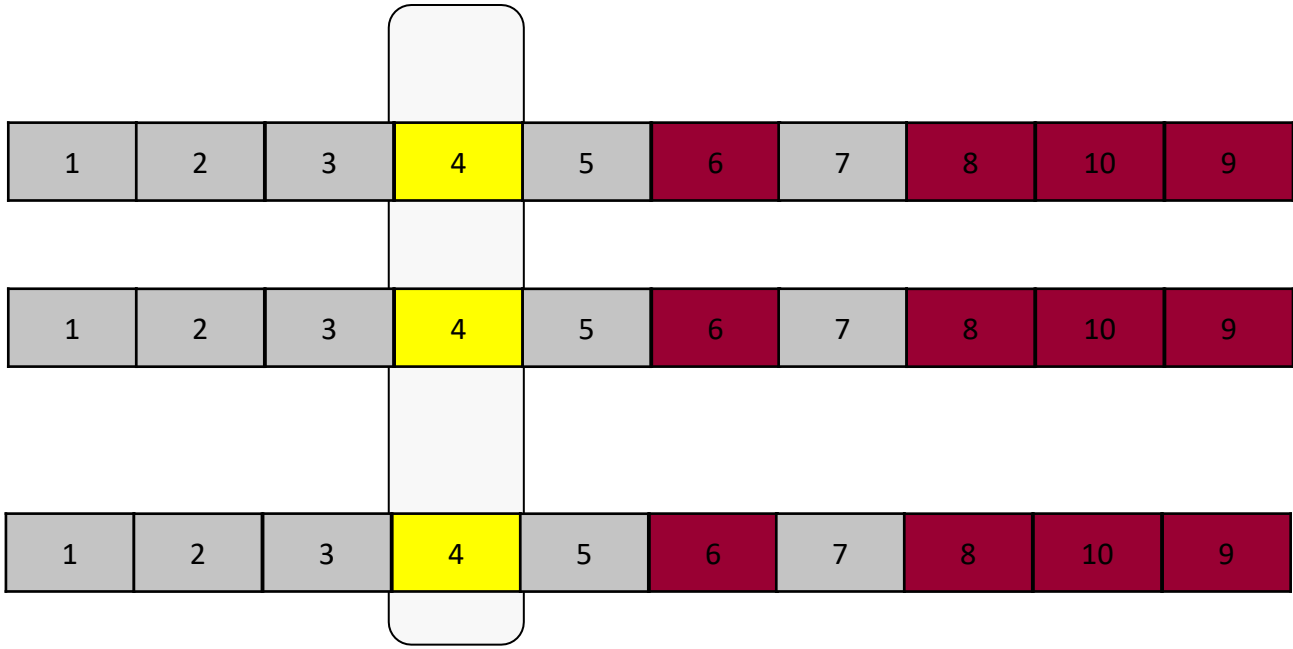


Swap with last element (4)

Partition around pivot

6th Call

Choose Random Pivot (4)



Swap with last element (4)

Partition around pivot

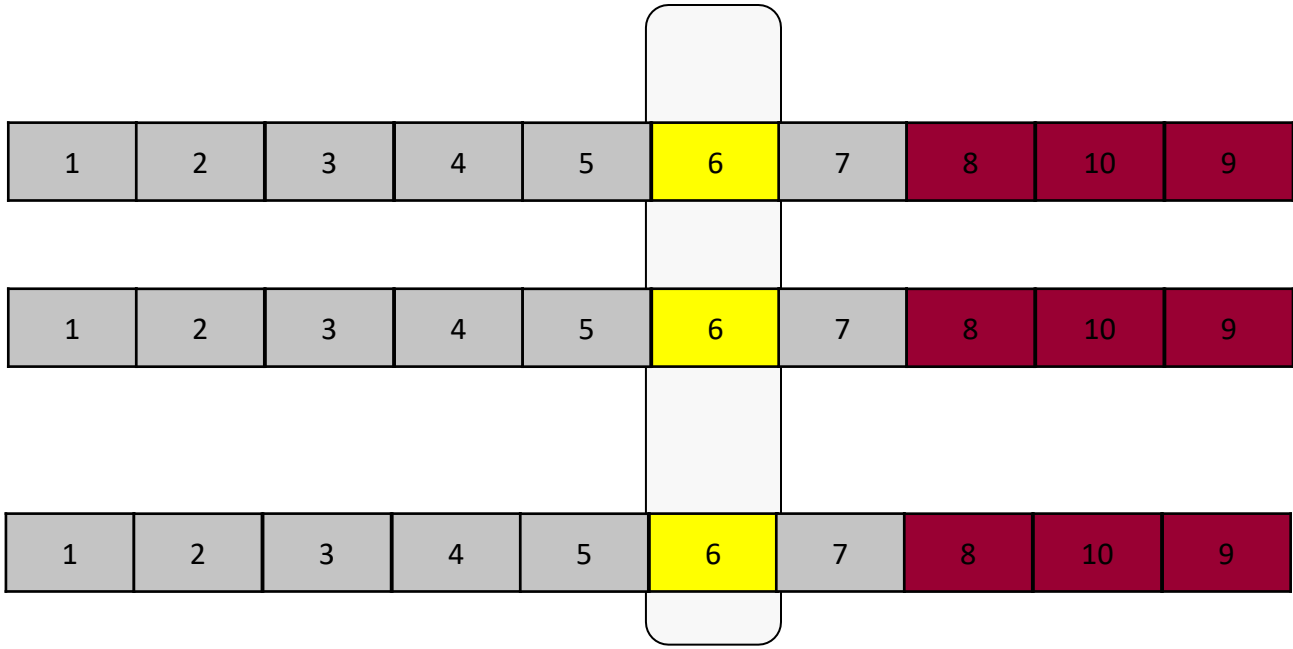
Since subarray contains only one element no work is actually done

7th Call

Choose Random Pivot (6)

Swap with last element (6)

Partition around pivot



Since subarray contains only one element no work is actually done

8th Call

Choose Random Pivot (10)

Swap with last element (9)

Partition around pivot

9th Call

Choose Random Pivot (8)

Swap with last element (9)

Partition around pivot

10th Call

Choose Random Pivot (9)

Swap with last element (9)

Partition around pivot

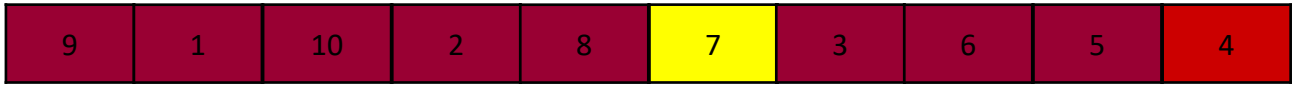
Since subarray contains only one element no work is actually done

Same thing again but with the corresponding pivot-tree

The tree is only constructed for purposes of the (randomized) analysis.
It is never actually drawn or used for the algorithm.

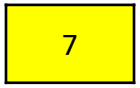
1st Call

Choose Random Pivot (7)



Swap with last element (4)

Partition around pivot

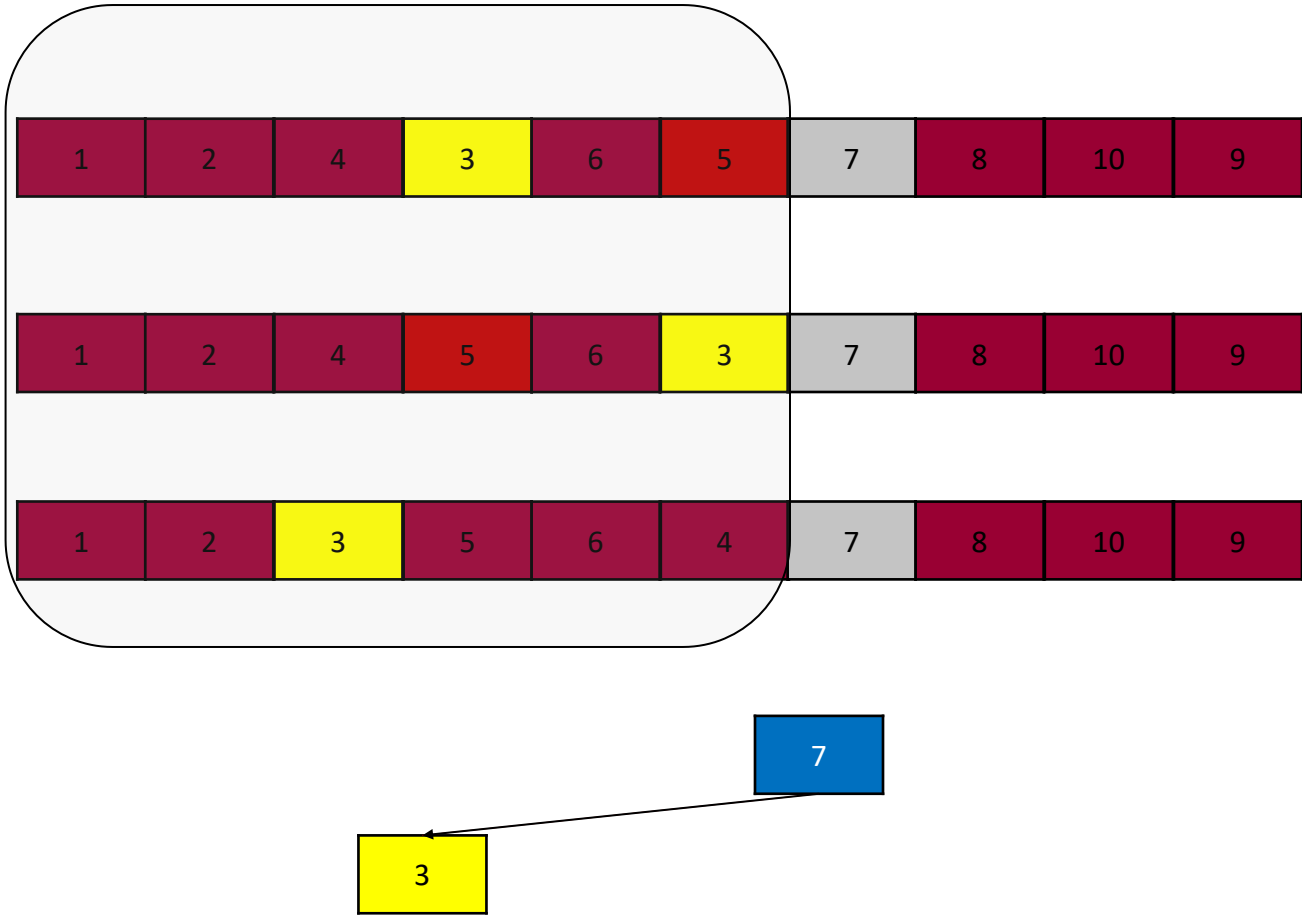


2nd Call

Choose Random Pivot (3)

Swap with last element (5)

Partition around pivot

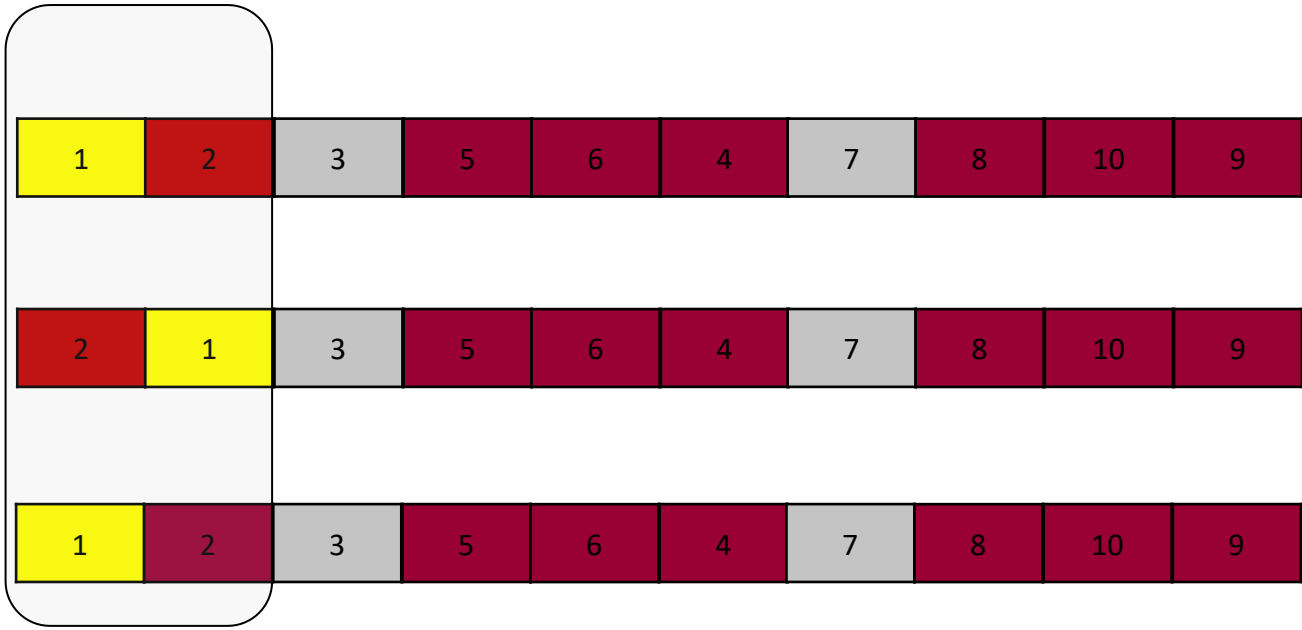


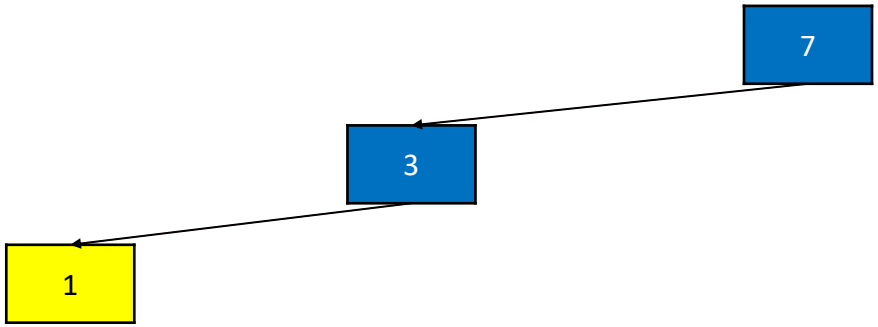
3rd Call

Choose Random Pivot (1)

Swap with last element (2)

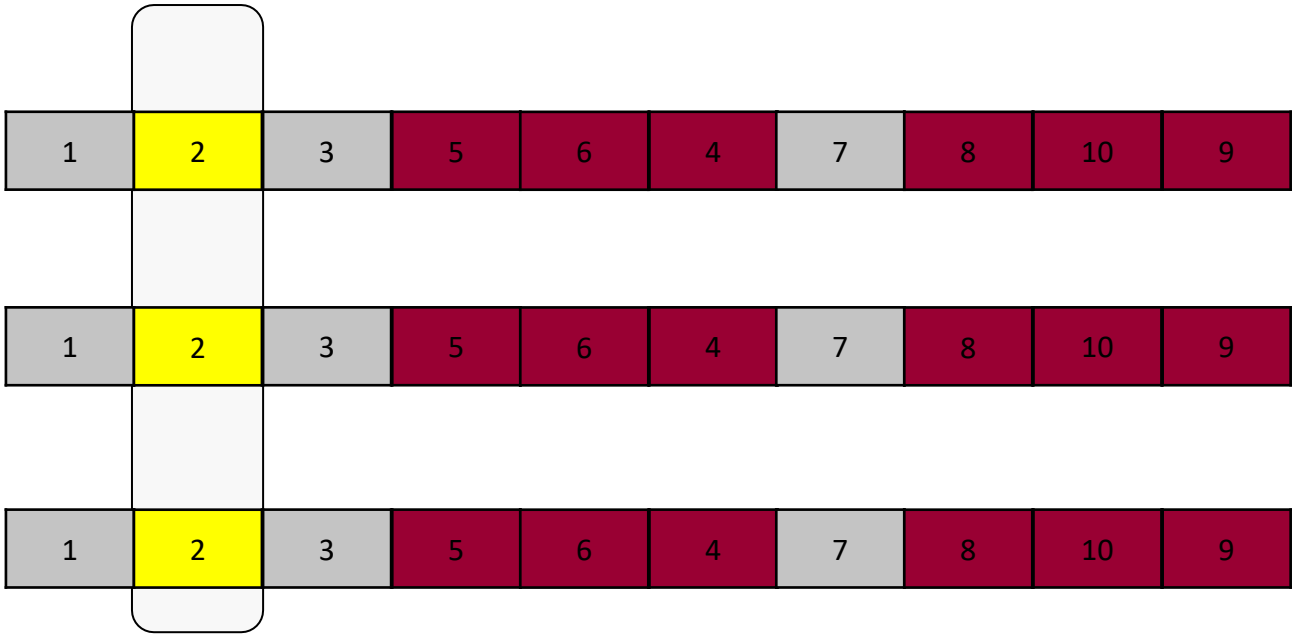
Partition around pivot





4th Call

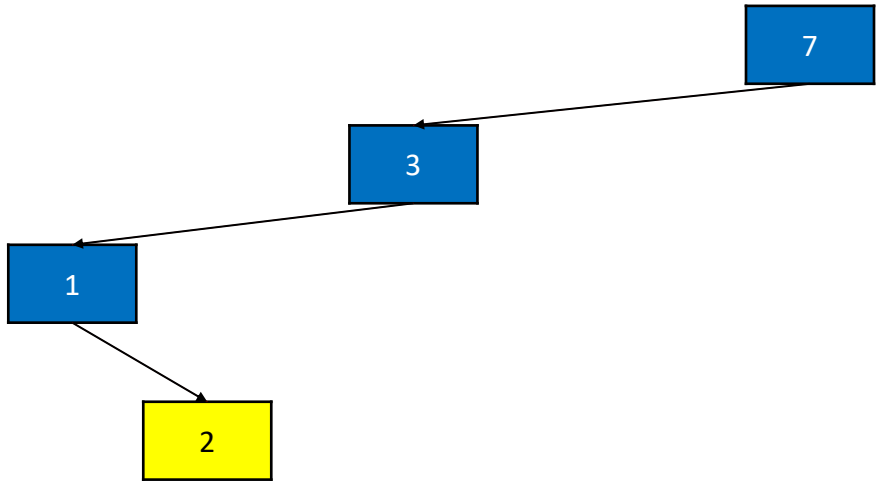
Choose Random Pivot (2)



Swap with last element (2)

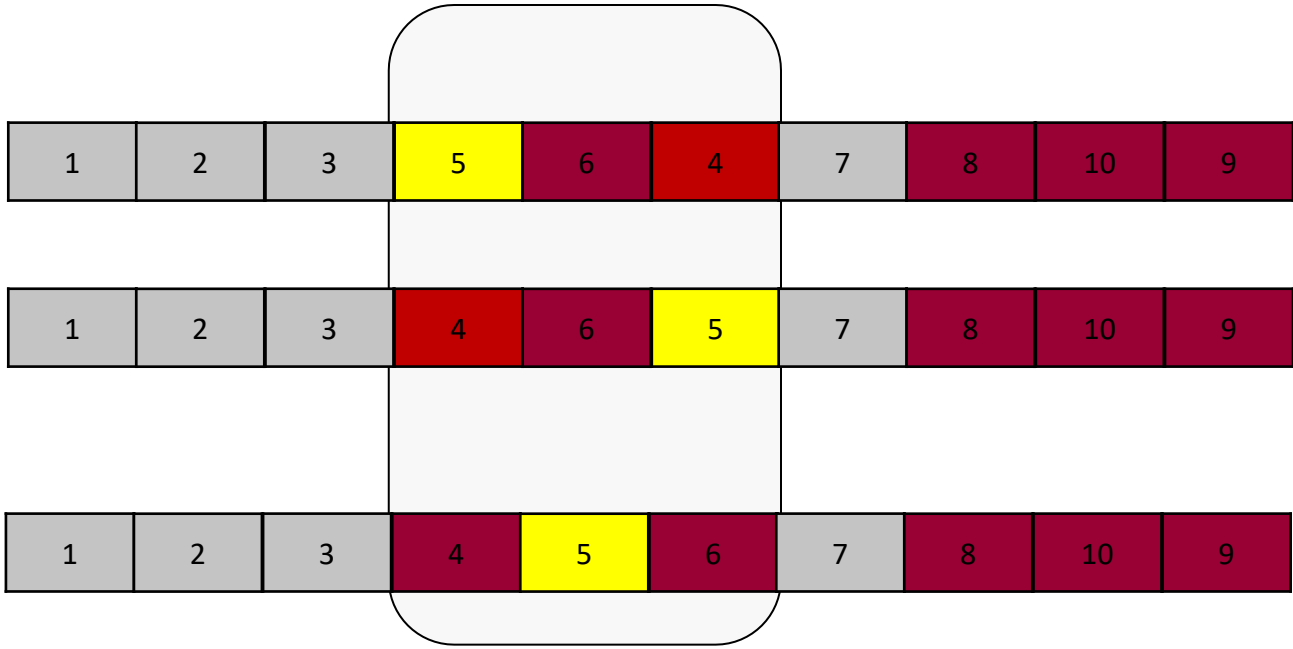
Partition around pivot

Since subarray contains only one element no work is actually done



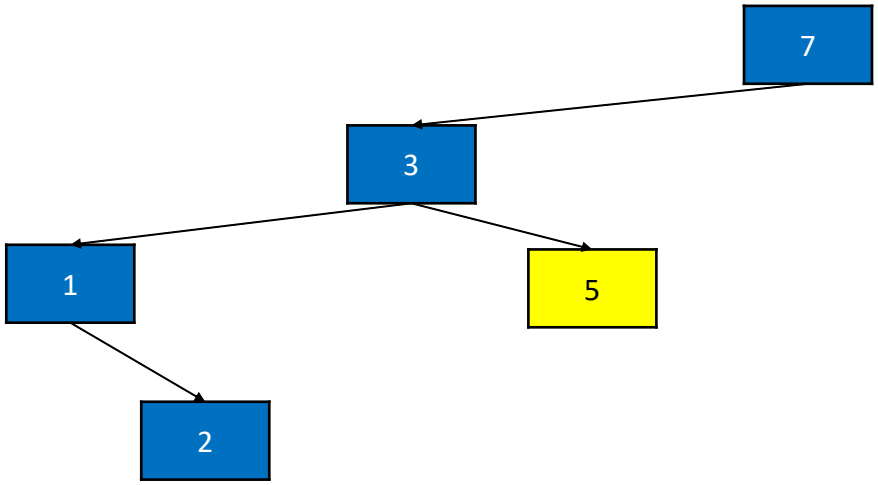
5th Call

Choose Random Pivot (5)



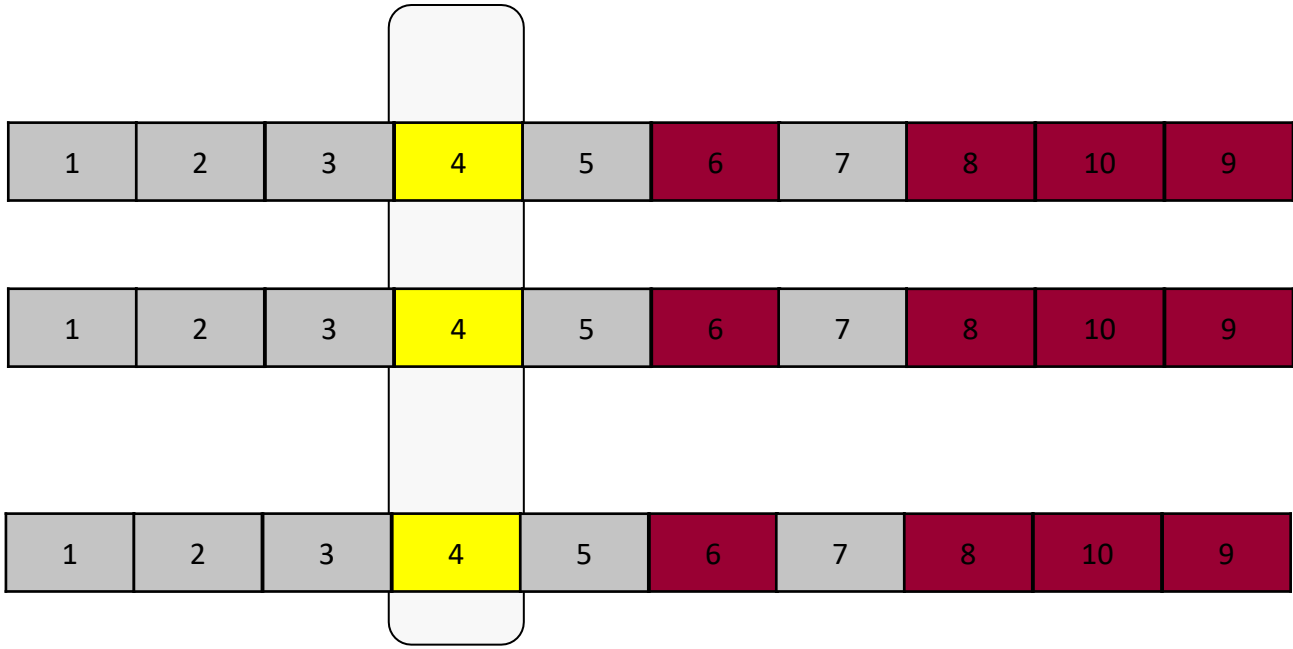
Swap with last element (4)

Partition around pivot



6th Call

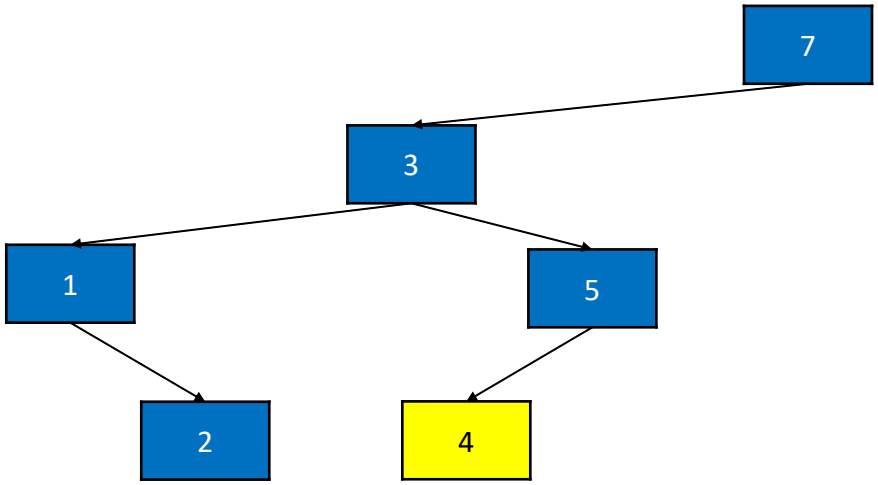
Choose Random Pivot (4)



Swap with last element (4)

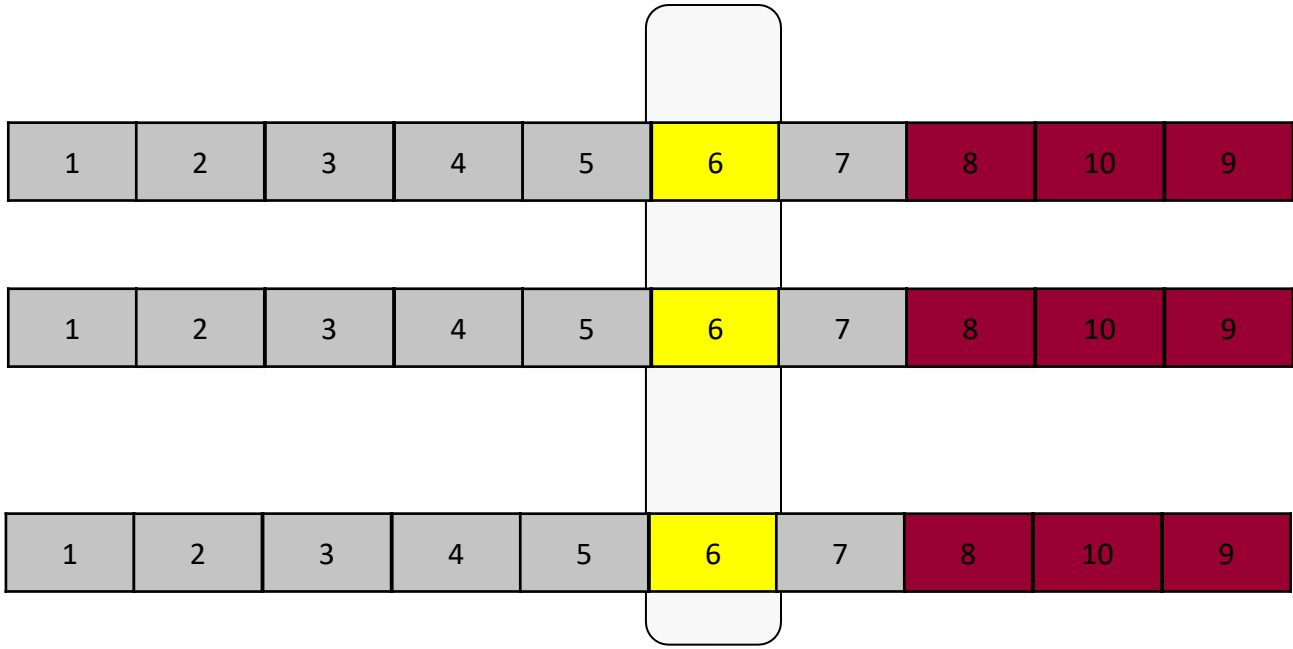
Partition around pivot

Since subarray contains only one element no work is actually done



7th Call

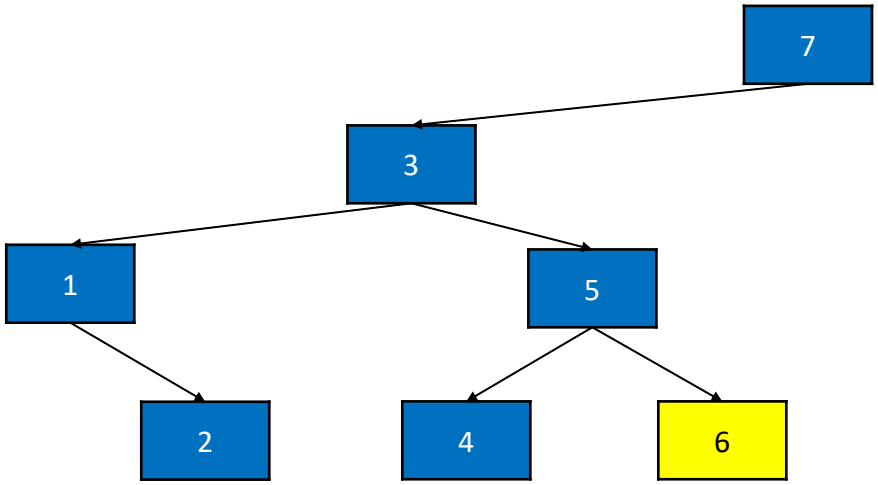
Choose Random Pivot (4)



Swap with last element (4)

Partition around pivot

Since subarray contains only one element no work is actually done

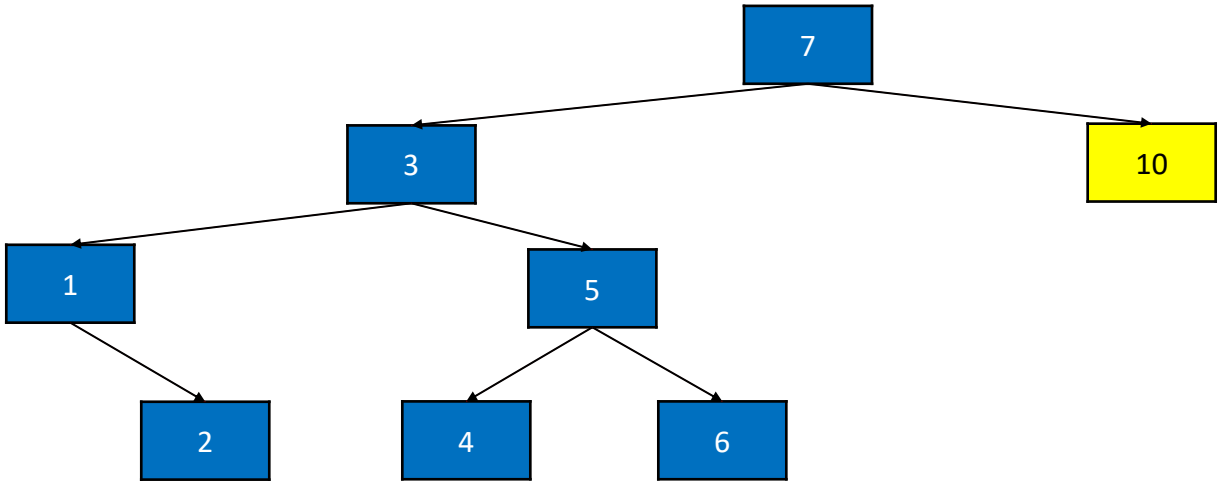


8th Call

Choose Random Pivot (10)

Swap with last element (9)

Partition around pivot

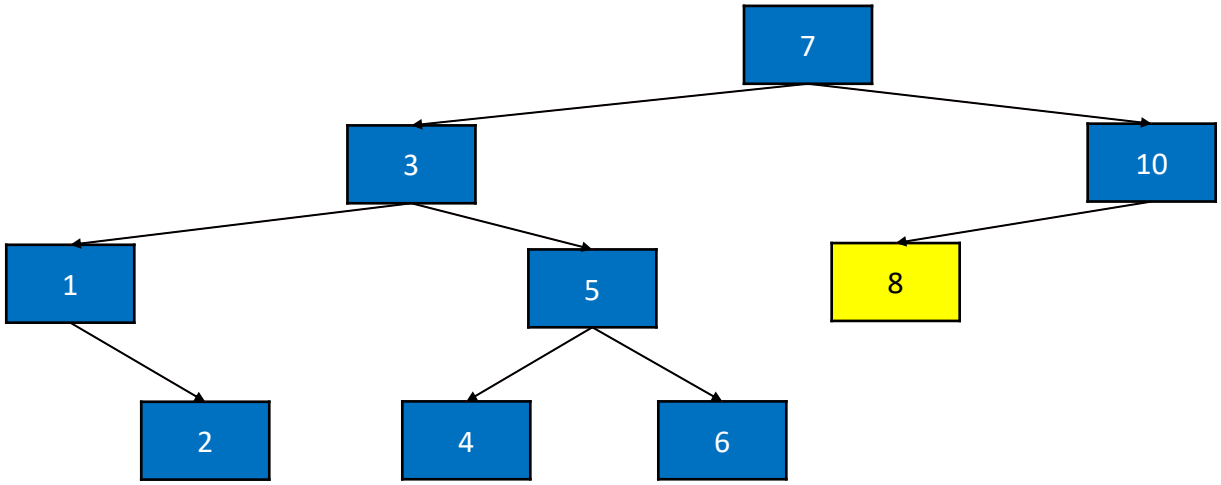


9th Call

Choose Random Pivot (8)

Swap with last element (9)

Partition around pivot



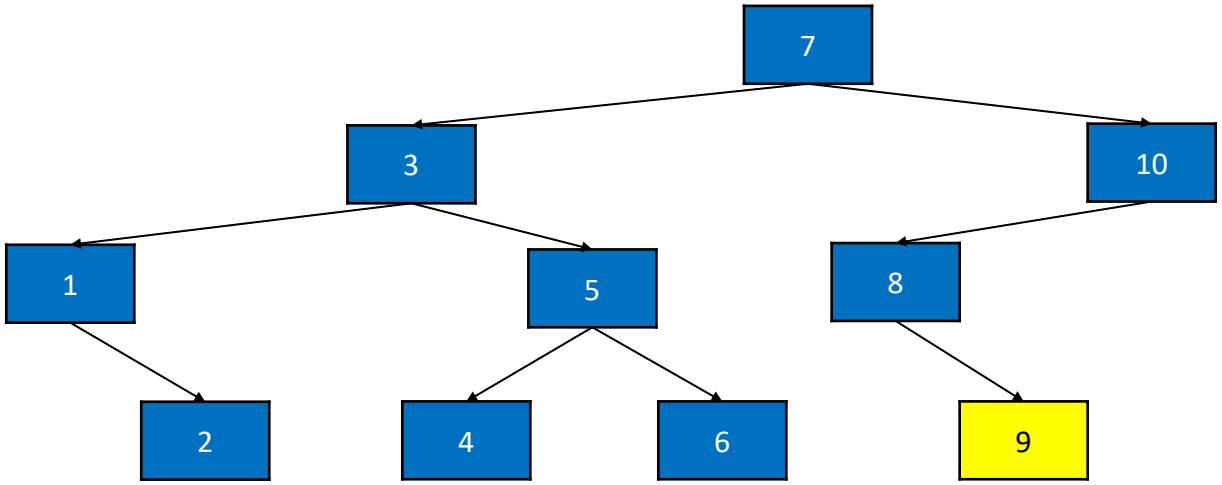
10th Call

Choose Random Pivot (9)

Swap with last element (9)

Partition around pivot

Since subarray contains only one element no work is actually done

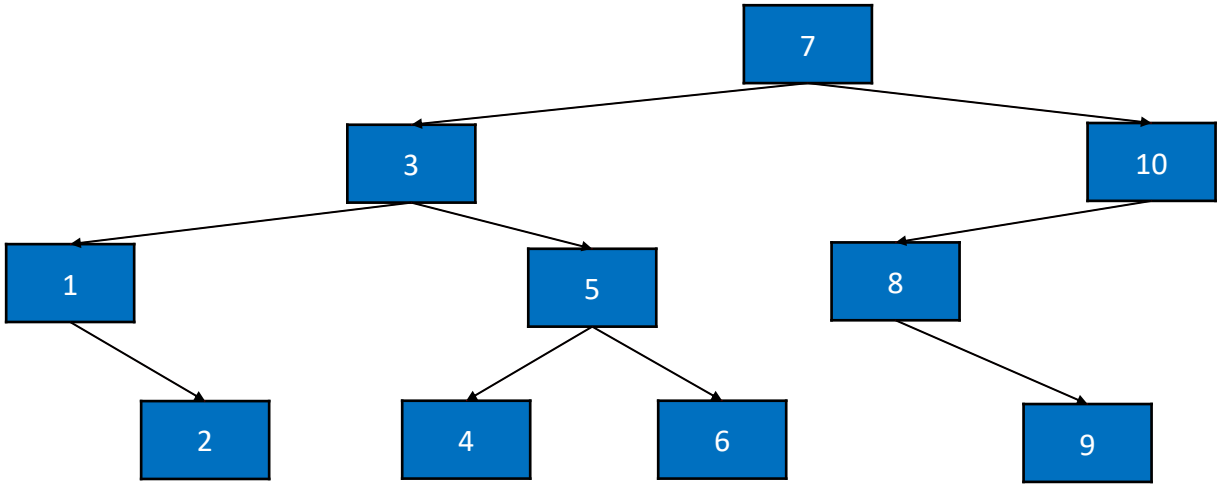


Review

Original Array

Quicksorted Array

Binary Tree Corresponding to Quicksort Pivot Selection



Entire algorithm execution can be reconstructed from tree