# UNIVERSITY INSTITUTE OF COMPUTING

## MASTER OF COMPUTER APPLICATIONS

### DESIGN AND ANALYSIS OF ALGORITHMS

### 24CAT-611

UNIT-2

DISCOVER . LEARN . EMPOWER

# Topics to be covered

- Divide and Conquer Method
- Binary Search
- Merge Sort
- Quick Sort

# Quick Sort

- Quick sort is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays.
- A large array is partitioned into two arrays one of which holds values smaller than the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value.
- Quicksort partitions an array and then calls itself recursively twice to sort the two resulting subarrays.
- This algorithm is quite efficient for large-sized data sets as its average and worst-case complexity are O(nLogn) and O($n^2$), respectively.

# How Quick Sort Works?

Following are the steps involved in quick sort algorithm:

1. After selecting an element as **pivot**, which is the last index of the array in our case, we divide the array for the first time.
2. In quick sort, we call this **partitioning**. It is not simple breaking down of array into 2 subarrays, but in case of partitioning, the array elements are so positioned that all the elements smaller than the **pivot** will be on the left side of the pivot and all the elements greater than the pivot will be on the right side of it.
3. And the **pivot** element will be at its final **sorted** position.
4. The elements to the left and right, may not be sorted.
5. Then we pick subarrays, elements on the left of **pivot** and elements on the right of **pivot**, and we perform **partitioning** on them by choosing a **pivot** in the subarrays.

# How Quick Sort Works?

1. A pivot element is chosen from the array. You can choose any element from the array as the pivot element.
Here, we have taken the rightmost (ie. the last element) of the array as the pivot element.

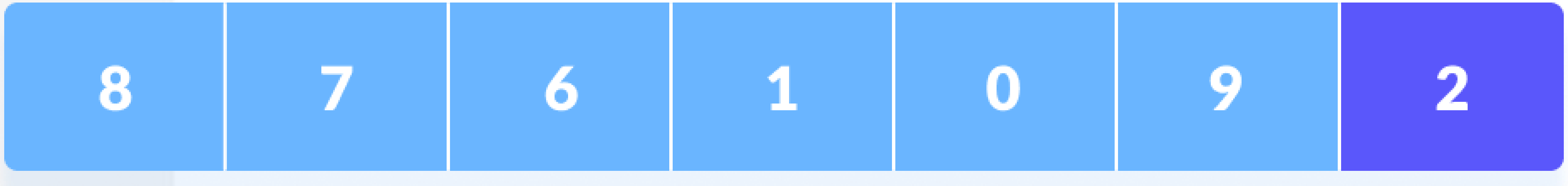
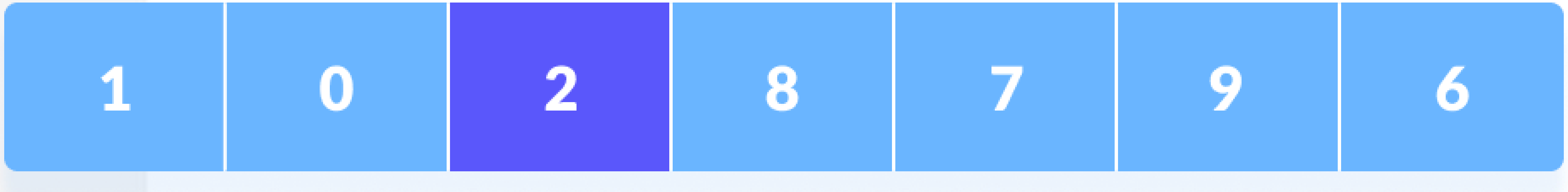


Figure1: Choosing Pivot element

# Contd…

2. The elements smaller than the pivot element are put on the left and the elements greater than the pivot element are put on the right.



| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 8 | 7 | 9 | 6 |

Figure 2: Setting pivot element

# Contd…

The above arrangement is achieved by the following steps.

- A pointer is fixed at the pivot element. The pivot element is compared with the elements beginning from the first index. If the element greater than the pivot element is reached, a second pointer is set for that element.
- Now, the pivot element is compared with the other elements (a third pointer). If an element smaller than the pivot element is reached, the smaller element is swapped with the greater element found earlier.

# Contd…

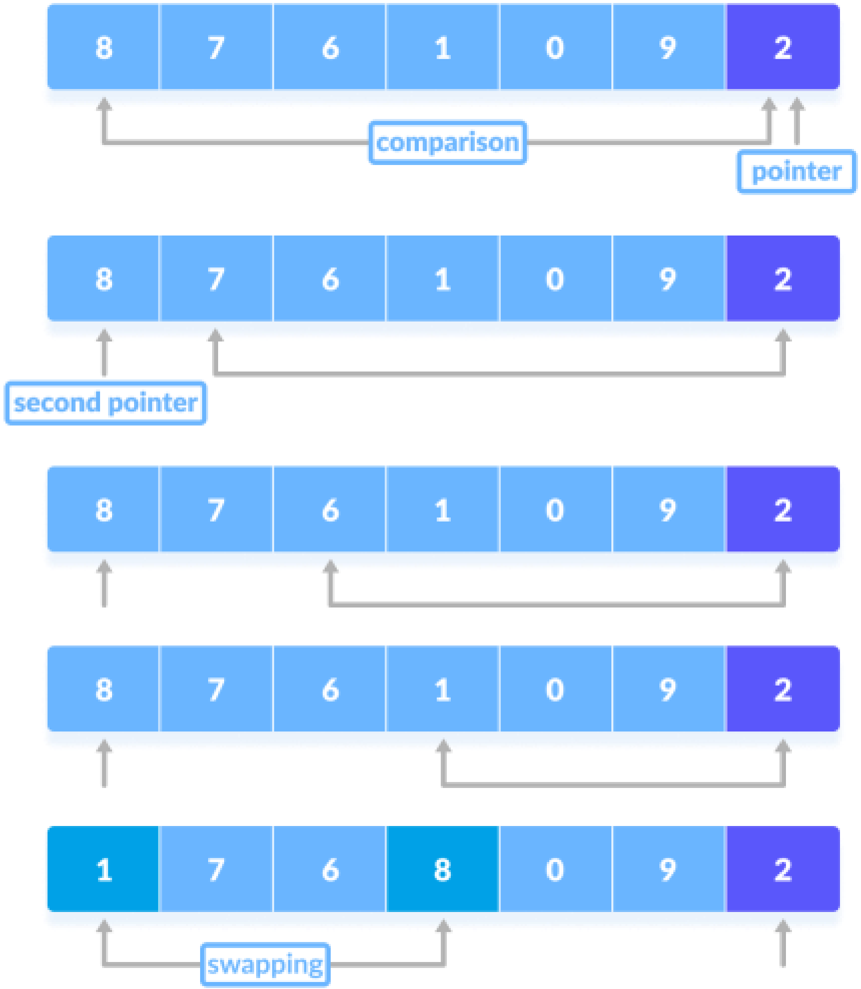


Figure3: Swapping

# Contd…

The process goes on until the second last element is reached.
Finally, the pivot element is swapped with the second pointer.

# Contd…

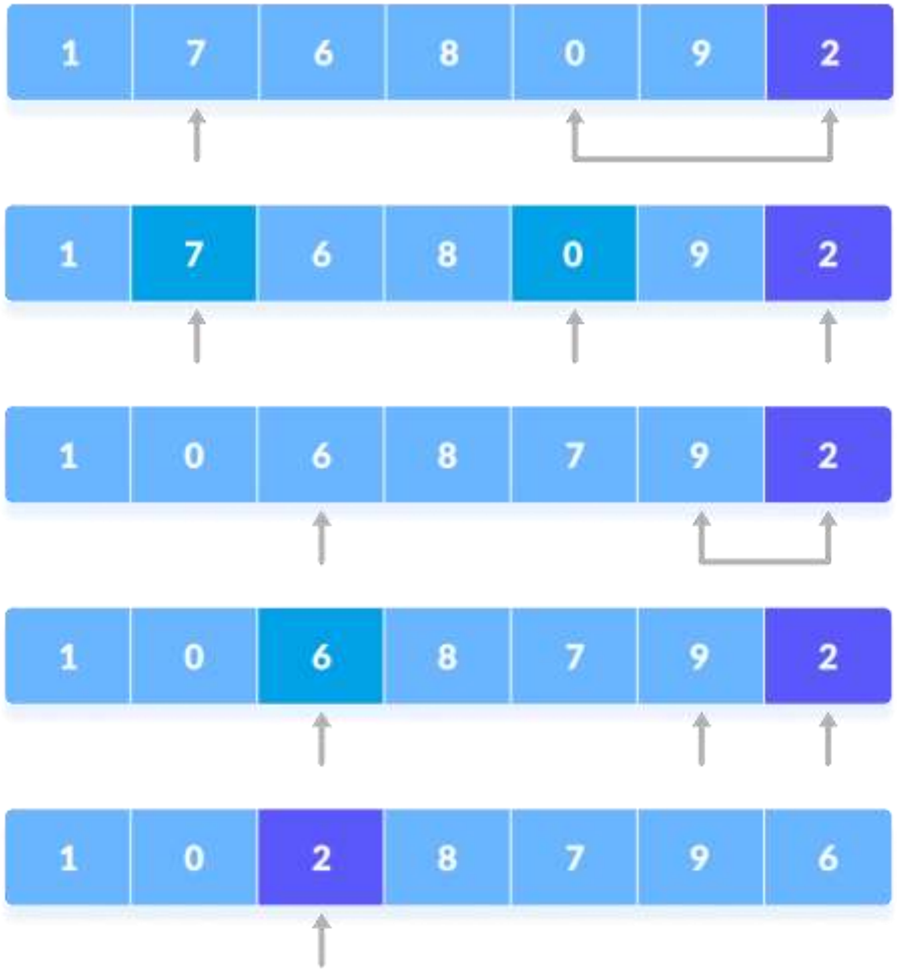


Figure 4: Swapping

# Contd…

3. Pivot elements are again chosen for the left and the right sub-parts separately. Within these sub-parts, the pivot elements are placed at their right position. Then, step 2 is repeated.

# Contd…

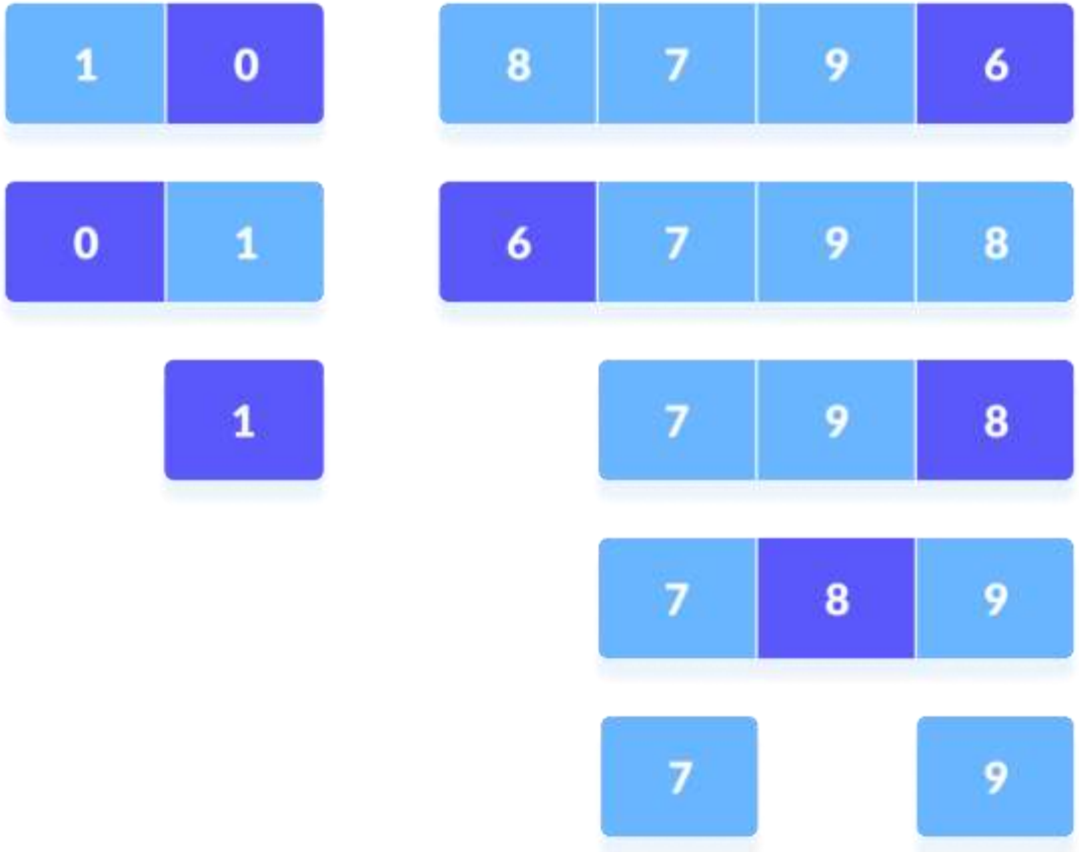


Figure 5: Partitioning

# Contd…

4. The sub-parts are again divided into smaller sub-parts until each subpart is formed of a single element.

5. At this point, the array is already sorted.

# Picking Pivot Element

- Like Merge Sort, QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.
- Always pick first element as pivot.
- Always pick last element as pivot (implemented below)
- Pick a random element as pivot.
- Pick median as pivot.

# Frequently asked questions

- What do you understand by quick sort?
- Define complexity of quick sort.

# References

[1] https://cdn.programiz.com/sites/tutorial2program/files/quick-sort-0.1_0.png

[2] https://cdn.programiz.com/sites/tutorial2program/files/quick-sort-0.2_0.png

[3] https://cdn.programiz.com/sites/tutorial2program/files/quick-sort-partition_1.png

[4] https://cdn.programiz.com/sites/tutorial2program/files/quick-sort-partition_1.png

[5] https://cdn.programiz.com/sites/tutorial2program/files/quick-sort-0.3_0.png

**Books Reference**

1. Introduction to Algorithms by Coreman, Leiserson, Rivest, Stein.
2. Fundamentals of Algorithms by Ellis Horwitz, Sartaj Sahni, Sanguthevar Rajasekaran

NAAC GRADE A+

# THANK YOU