Design and Analysis of Algorithms

Name of the Faculty: Mr.M.Raja Babu, Assistant Professor, Department of IT
Subject: Design and Analysis of Algorithm
Year & Semester: II -II
Topic: Quick Sort
Conventional Methods: Chalk & Talk
Teaching Methodology: Real-time case studies

In Design and Analysis of Algorithm subject, Quick Sort technique is an important topic. Many have taught the topic using chalk and talk. Mr. M. Rajababu has taught the topic of quicksort using real-time case studies. Students can understand the practical applications of quicksort through real time case studies. If the topic explained without any case study, then student was not able to understand how to use quicksort in real time. So, the innovative method of real time case studies used for explaining real time applications of quicksort.

References:

1.https://www.javatpoint.com/quick-sort

2. https://www.tutorialspoint.com/data_structures_algorithms/quick_sort_algorithm.htm

3. https://www.youtube.com/watch?v=gtWw_8VvHjk

Quick sort

It is used on the principle of divide-and-conquer. Quick sort is an algorithm of choice in many situations as it is not difficult to implement. It is a good general purpose sort and it consumes relatively fewer resources during execution.

Advantages

- It is in-place since it uses only a small auxiliary stack.
- It requires only n (log n) time to sort n items.
- It has an extremely short inner loop.
- This algorithm has been subjected to a thorough mathematical analysis, a very precise statement can be made about performance issues.

Disadvantages

- It is recursive. Especially, if recursion is not available, the implementation is extremely complicated.
- It requires quadratic (i.e., n2) time in the worst-case.
- It is fragile, i.e. a simple mistake in the implementation can go unnoticed and cause it to perform badly.

Quick sort works by partitioning a given array A[p ... r] into two non-empty sub array A[p ... q] and A[q+1 ... r] such that every key in A[p ... q] is less than or equal to every key in A[q+1 ... r].
Then, the two sub-arrays are sorted by recursive calls to Quick sort. The exact position of the partition depends on the given array and index q is computed as a part of the partitioning procedure.
Algorithm: Quick-Sort (A, p, r)

```
if p < r then
   q Partition (A, p, r)
   Quick-Sort (A, p, q)
   Quick-Sort (A, q + r, r)
```

Note that to sort the entire array, the initial call should be Quick-Sort (A, 1, length[A])
As a first step, Quick Sort chooses one of the items in the array to be sorted as pivot. Then, the array is partitioned on either side of the pivot. Elements that are less than or equal to pivot will move towards the left, while the elements that are greater than or equal to pivot will move towards the right.
Partitioning the Array
Partitioning procedure rearranges the sub-arrays in-place.
Function: Partition (A, p, r)

```
x ← A[p]
i ← p-1
j ← r+1
while TRUE do
   Repeat j ← j - 1
   until A[j] ≤ x
   Repeat i← i+1
```

```
  until A[i] ≥ x
  if i < j then
    exchange A[i] ↔ A[j]
  else
    return j
```

Analysis

The worst case complexity of Quick-Sort algorithm is $O(n^2)$. However using this technique, in average cases generally we get the output in O(n log n) time.

Real-time application of Quicksort:

Commercial Computing is used in various government and private organizations for the purpose of sorting various data like sorting files by name/date/price, sorting of students by their roll no., sorting of account profile by given id, etc.

The sorting algorithm is used for information searching and as Quicksort is the fastest algorithm so it is widely used as a better way of searching.

It is used everywhere where a stable sort is not needed.

Quicksort is a cache-friendly algorithm as it has a good locality of reference when used for arrays.

It is tail -recursive and hence all the call optimization can be done.

It is an in-place sort that does not require any extra storage memory.

It is used in operational research and event-driven simulation.

<u>Quick sort using real time case studies:</u>