CS435DE - Lab 5

Saurab Ghimire
417555

Problem 1: Solution

QuickSort Steps for Array [1, 6, 2, 4, 3, 5] using Leftmost Pivot:

Step 1:
Pivot = Left element = 1.
After partitions, we get: [1] + [6, 2, 4, 3, 5]

Step 2:
Array = [6, 2, 4, 3, 5],
Pivot = 6.
After partitions, we get: [2, 4, 3, 5] + [6]

Step 3:
Array = [2, 4, 3, 5],
Pivot = 2.
After partitions, we get: [2] + [4, 3, 5]

Step 4:
Array = [4, 3, 5],
Pivot = 4.
After partitions, we get: [3] + [4] + [5]

Step 5:
Finally, the array is sorted: [1, 2, 3, 4, 5, 6].

Problem 3: Solution

The algorithm should be as follows:

Step 1: Using the sorted array, make use of binary search

Step 2: Compare the middle element with its index:

  If $A[m] = m$, return m.

  If $A[m] > m$, search the left half of the array.

  If $A[m] < m$, search the right half of the array.

This takes $O(\log n)$ time, which is less than $O(n)$.

Proof:

With each comparison, binary search divides the array into two partitions, and this process continues for log times, making the time complexity $O(\log n)$. Since $\log n$ grows much slower than n, this algorithm runs in $o(n)$ time.

Problem 4: Solution

Select the pivot as the "median of medians." This guarantees that the pivot is a good choice, providing balanced partitions.

This algorithm guarantees $O(n)$ time for selecting the pivot and ensures that the QuickSort will have a worst-case time complexity of $O(n \log n)$.

Problem 5: Solution

The Array that is provided: [1, 12, 8, 7, -2, -3, 6] (n = 7, median is the 4th element).
Step 1: Pivot = 1 (leftmost element).
Partition: [-3, -2] | 1 | [12, 8, 7, 6]. Since the median is in the right part, continue with [12, 8, 7, 6].
Step 2: Pivot = 12 (leftmost element).
Partition: [8, 7, 6] | 12 |. The median is in the left part, continue with [8, 7, 6].
Step 3: Pivot = 8 (leftmost element).
Partition: [7, 6] | 8 | . The median is 6 (the 4th element in the array).
The median is 6.