National Institute of Technology Calicut
Department of Computer Science and Engineering
Third Semester B. Tech.(CSE)
CS2092D Programming Laboratory
Assignment #3 - Practice Question

QUESTION

General Note: For the following program, do not declare the array as a global variable. You may pass the array as a function argument using the concept of pointers.

1. Write a program that uses the QUICK-SORT algorithm for sorting a given input sequence of integers present in an array A and prints the number of comparisons performed during sorting. Your program must contain the following functions: (the notation $A[i..j]$ denotes the sub-array of A, contained within the i^{th} and j^{th} indices, both inclusive)

   - A recursive function QUICK-SORT($A,\ p,\ r$) that takes as input an array A and sorts the sub-array $A[p..r]$ using Quick-Sort.
   - A function PARTITION($A,\ p,\ q,\ r$) that takes as input an array A and partitions it into two sub arrays $A[p..q-1]$ and $A[q+1..r]$ such that each element of $A[p..q-1]$ is less than or equal to $A[q]$ which is, in turn, less than or equal to each element of $A[q+1..r]$.
   - PRINT($A,\ n$) - A function that takes as input an array A and an integer n, the size of A. It then prints the contents of A in order, with a single space separating the elements. This function should only be called from the MAIN() function.

   Input format:

   - The first line of the input contains an integer $n \in [0, 10^5]$, the size of the array A.
   - The second line lists the n elements in A, as space-separated integers in the range $[-10^3, 10^3]$.

   Output Format:

   - The first line of the output contains the elements of A in sorted order, separated by a space.
   - The second line of the output contains the number of comparisons performed during sorting.

   Note:

   The number of comparisons made by Quick-Sort is highly dependent on its implementation. As such, we will be considering the number of comparisons as per the algorithm given in CLRS.

   Sample Input 1:

   ```
   7
   1 2 3 4 5 6 7
   ```

   Sample Output 1:

   ```
   1 2 3 4 5 6 7
   21
   ```

   Sample Input 2:

   ```
   7
   1 2 5 7 6 9 8
   ```

   Sample Output 2:

   ```
   1 2 5 6 7 8 9
   13
   ```