CSN-261: Data Structures Laboratory

Lab Assignment 1 (L1)

Instructions:

1) **Use either C or C++ for solving the assignment.**
2) **Throughout the assignment, *n* represents the number of input.**
3) **Array index starts with 0 in C and C++.**

Problem 1. Write a program to implement the Quick sort algorithm with expected-time complexity of O(nlogn).
Test Case [**Input:** 50 23 1 5 10 4 8 3 **Output:** 1 3 4 5 8 10 23 50]

Problem 2. Write a program to search the index of the element from a sorted array with O(log n) worst-case time complexity.
Test Case [**Input:** 1 3 4 5 8 10 23 50 and search **Key** value = 8, **Output:** 4]

Problem 3. Write a program that reverses an array of integer in-place. No variable can be used in the program apart from loop variables.
Test Case **[Input:** 5 28 2 1 4 12 **Output:** 12 4 1 2 28 5**]**

Problem 4. Write a program to implement bit-level Radix sort. Accept *n* random integers (in decimal) as input and convert them into 8-bit binary (Keep the range of input integers between 0-255). Consider the decimal value of 2 bits at a time from the least significant bit (LSB) to the most significant bit (MSB) of the 8-bit binary to perform the sorting operation. Print the output in the form of decimal values.
(Hint: *Use count sort***)**
Test Case [**Input:** 50 23 1 5 10 4 8 3 **Output:** 1 3 4 5 8 10 23 50]

For Example:
9 => 00 00 10 01->1
3 => 00 00 00 11->3
10 => 00 00 10 10->2

Problem 5. Write a program to implement the Insertion sort algorithm with worst-case time complexity of O(nlogn) for the number of comparisons performed. Use the concept of binary search in classical Insertion sort while searching the element for the insertion in each iteration. The worst-case comparisons for this approach will be O(nlogn) but the overall complexity will be O(n^2) because the number of swaps will remain O(n^2).
Test Case [**Input:** 50 23 1 5 10 4 8 3 **Output:** 1 3 4 5 8 10 23 50]