Name : Pratik Rajesh Jade

Roll no : A 70

Aim - Sort a given set of n integer elements using Quick Sort method and compute its time

complexity. Run the program for varied values of n and record the time taken to sort. The

elements can be read from a user or can be generated using the random number generator.

Demonstrate using C++ how the divide and conquer method works along with its time

complexity analysis: worst case, average case and best case.

#include <iostream>

#include <ctime>

void quickSort(int arr[], int left, int right)

{

    int i = left, j = right;

    int pivot = arr[(left + right) / 2];

    while (i <= j)

    {

        while (arr[i] < pivot)

            i++;

        while (arr[j] > pivot)

            j--;

        if (i <= j)

        {

            int temp = arr[i];

            arr[i] = arr[j];

            arr[j] = temp;

            i++;

            j--;

        }

    }

    if (left < j)

        quickSort(arr, left, j);

    if (i < right)

        quickSort(arr, i, right);

}

int main()

{

    int n;

    std::cout << "Enter the number of elements: ";

    std::cin >> n;

    int \*arr = new int[n];

    srand(time(0));

    for (int i = 0; i < n; i++)

        arr[i] = rand() % 1000 + 1;

    clock\_t start = clock();

    quickSort(arr, 0, n - 1);

    clock\_t stop = clock();

    double duration = (double)(stop - start) / CLOCKS\_PER\_SEC;

    std::cout << "Sorted array: ";

    for (int i = 0; i < n; i++)

        std::cout << arr[i] << " ";

    std::cout << std::endl;

    std::cout << "Time taken by function: " << duration << " seconds" << std::endl;

    return 0;

}

Output:

