**Merge Sort**

Design and implement C/C++ Program to sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

// Merge two subarrays of arr[]

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

// Create temporary arrays

int L[n1], R[n2];

// Copy data to temporary arrays L[] and R[]

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

R[j] = arr[m + 1 + j];

// Merge the temporary arrays back into arr[l..r]

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

// Copy the remaining elements of L[], if any

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

// Copy the remaining elements of R[], if any

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

// Merge sort function

void mergeSort(int arr[], int l, int r)

{

if (l < r)

{

// Same as (l+r)/2, but avoids overflow for large l and r

int m = l + (r - l) / 2;

// Sort first and second halves

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

// Merge the sorted halves

merge(arr, l, m, r);

}

}

int main()

{

int n;

printf("Enter the number of elements: ");

scanf("%d", &n);

int arr[n];

// Generate random numbers for the array

srand(time(NULL));

for (int i = 0; i < n; i++)

{

arr[i] = rand() % 10000; // You can adjust the range of random numbers as needed

}

// Measure the time taken to sort the array

clock\_t start, end;

double cpu\_time\_used;

start = clock();

mergeSort(arr, 0, n - 1);

end = clock();

cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;

printf("Sorted array:\n");

for (int i = 0; i < n; i++)

{

printf("%d ", arr[i]);

}

printf("\n");

printf("Time taken for sorting: %f seconds\n", cpu\_time\_used);

return 0;

}