



Design and Analysis of Algorithms - Lab
Experiment 2

NAME: Ritik Kumar

SAP ID: 590017256

PROGRAM: B.Tech CSE

BATCH: 34

SUBMITTED TO: Mr. Aryan Gupta

GitHub Repository Link:

https://github.com/Ritik2807/DAA_LAB_2_Ritik_Kumar_590017256.git

Objective: Implement merge sort using any of the programming language (C, C++, Java). Make one pdf, including code, 10 different test cases and their screenshots. Also attach plagiarism report in the end.

Source Code (C):

```
#include <stdio.h>
```

```
void merge(int arr[], int low, int mid, int high)
```

```
{
```

```
    int n1 = mid - low + 1;
```

```
    int n2 = high - mid;
```

```
    int Left[n1], Right[n2];
```

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

```
        Left[i] = arr[low + i];
```

```
    for (int i = 0; i < n2; i++)
```

```
        Right[i] = arr[mid + 1 + i];
```

```
    int left = 0, right = 0, k = low;
```

```
    while (left < n1 && right < n2)
```

```
    {
```

```
        if (Left[left] <= Right[right])
```

```
        {
```

```
            arr[k] = Left[left];
```

```
            left++;
```

```
            k++;
```

```
        }
```

```
    else
```

```
    {
```

```
        arr[k] = Right[right];
```

```
        k++;
```

```
        right++;
```

```
    }
```

```
}
```

```
while (left < n1)
```

```
    arr[k++] = Left[left++];
```

```
while (right < n2)
```

```

        arr[k++] = Right[right++];
    }

void mergeSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int mid = low + (high - low) / 2;

        mergeSort(arr, low, mid);
        mergeSort(arr, mid + 1, high);

        merge(arr, low, mid, high);
    }
}

void printArray(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main()
{
    int n;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    int low = 0, high = n - 1;

    printf("Original Array: ");
    printArray(arr, n);

    mergeSort(arr, low, high);

    printf("Sorted Array: ");
    printArray(arr, n);

```

return 0;

}

DAA_LAB_Ritik_Kumar_590017256 > C merge_sort.c > main()

```
1  #include <stdio.h>
2
3  void merge(int arr[], int low, int mid, int high)
4  {
5      int n1 = mid - low + 1;
6      int n2 = high - mid;
7
8      int Left[n1], Right[n2];
9      for (int i = 0; i < n1; i++)
10         Left[i] = arr[low + i];
11     for (int i = 0; i < n2; i++)
12         Right[i] = arr[mid + 1 + i];
13     int left = 0, right = 0, k = low;
14
15     while (left < n1 && right < n2)
16     {
17         if (Left[left] <= Right[right])
18         {
19             arr[k] = Left[left];
20             left++;
21             k++;
22         }
23         else
24         {
25             arr[k] = Right[right];
26             k++;
27             right++;
28         }
29     }
30     while (left < n1)
31         arr[k++] = Left[left++];
32     while (right < n2)
33         arr[k++] = Right[right++];
34 }
35
36 void mergeSort(int arr[], int low, int high)
37 {
38     if (low < high)
39     {
40         int mid = low + (high - low) / 2;
41
42         mergeSort(arr, low, mid);
43         mergeSort(arr, mid + 1, high);
44
45         merge(arr, low, mid, high);
46     }
47 }
48 void printArray(int arr[], int n)
49 {
50     for (int i = 0; i < n; i++)
51         printf("%d ", arr[i]);
52     printf("\n");
53 }
54 int main()
55 {
56     int n;
57
58     printf("Enter number of elements: ");
59     scanf("%d", &n);
60
61     int arr[n];
62     printf("Enter %d elements:\n", n);
63     for (int i = 0; i < n; i++)
64         scanf("%d", &arr[i]);
65
66     int low = 0, high = n - 1;
67
68     printf("Original Array: ");
69     printArray(arr, n);
70
71     mergeSort(arr, low, high);
72
73     printf("Sorted Array: ");
74     printArray(arr, n);
75
76     return 0;
77 }
```

All 10 Test cases:

Test Case 1 (Random numbers):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] [ ] X

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 6
Enter 6 elements:
12 11 13 5 6 7
Original Array: 12 11 13 5 6 7
Sorted Array: 5 6 7 11 12 13
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 2 (Small array, shuffled):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] [ ] X

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 3
Enter 3 elements:
3 1 2
Original Array: 3 1 2
Sorted Array: 1 2 3
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 3 (Reverse sorted array):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] [ ] X

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 6
Enter 6 elements:
10 9 8 7 6 5
Original Array: 10 9 8 7 6 5
Sorted Array: 5 6 7 8 9 10
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 4 (Single element):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] [ ] X

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 1
Enter 1 elements:
1
Original Array: 1
Sorted Array: 1
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 5 (All equal elements):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] [ ] X

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 4
Enter 4 elements:
2 2 2 2
Original Array: 2 2 2 2
Sorted Array: 2 2 2 2
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 6 (Mix of positive and negative):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] x

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 5
Enter 5 elements:
100 -10 0 50 20
Original Array: 100 -10 0 50 20
Sorted Array: -10 0 20 50 100
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 7 (More elements, shuffled):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] x

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 8
Enter 8 elements:
9 8 3 7 5 2 6 1
Original Array: 9 8 3 7 5 2 6 1
Sorted Array: 1 2 3 5 6 7 8 9
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 8 (Descending order):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] x

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 5
Enter 5 elements:
5 4 3 2 1
Original Array: 5 4 3 2 1
Sorted Array: 1 2 3 4 5
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 9 (All zeros):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] x

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 5
Enter 5 elements:
0 0 0 0 0
Original Array: 0 0 0 0 0
Sorted Array: 0 0 0 0 0
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```

Test Case 10 (Mixed numbers, random):

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - DAA_LAB_Ritik_Kumar_590017256 + - [ ] ... [ ] x

PS C:\Users\91898\OneDrive\Desktop\C Programming> cd "c:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256\" ; if ($?) { gcc merge_sort.c -o merge_sort } ;
if ($?) { .\merge_sort }
Enter number of elements: 7
Enter 7 elements:
15 23 1 89 55 68 31
Original Array: 15 23 1 89 55 68 31
Sorted Array: 1 15 23 31 55 68 89
PS C:\Users\91898\OneDrive\Desktop\C Programming\DAA_LAB_Ritik_Kumar_590017256>
```