

WEEK 3

DAA LAB

Kavin.J.S
CH.SC.U4CSE24119

Merge Sort:

```
C mergesort.c X C quicksort.c
C: > Users > kavin > OneDrive > Desktop > DAA > WEEK - 3 > C mergesort.c >
1 #include <stdio.h>
2
3 #define MAX 1000
4
5 int temp[MAX];
6
7 void mergeSort(int a[], int low, int high);
8 void merge(int a[], int low, int mid, int high);
9
10 int main() {
11     int n, a[MAX];
12
13     printf("Enter number of elements: ");
14     scanf("%d", &n);
15
16     printf("Enter elements:\n");
17     for (int i = 0; i < n; i++)
18         scanf("%d", &a[i]);
19
20     mergeSort(a, 0, n - 1);
21
22     printf("Sorted array:\n");
23     for (int i = 0; i < n; i++)
24         printf("%d ", a[i]);
25
26     return 0;
27 }
28
29 void mergeSort(int a[], int low, int high) {
30     if (low < high) {
31         int mid = (low + high) / 2;
32
33         mergeSort(a, low, mid);
34         mergeSort(a, mid + 1, high);
35         merge(a, low, mid, high);
36     }
37 }
38
39 void merge(int a[], int low, int mid, int high) {
40     int i = low, j = mid + 1, k = low;
41
42     while (i <= mid && j <= high) {
43         if (a[i] <= a[j])
44             temp[k++] = a[i++];
45         else
46             temp[k++] = a[j++];
47     }
48
49     while (i <= mid)
50         temp[k++] = a[i++];
51
52     while (j <= high)
53         temp[k++] = a[j++];
54
55     for (i = low; i <= high; i++)
56         a[i] = temp[i];
57 }

C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>gcc mergesort.c
C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>a
Enter number of elements: 6
Enter elements:
0 1 2 7 8 5
Sorted array:
0 1 2 5 7 8
```

Quick Sort:

```
C mergesort.c C quicksort.c X
C: > Users > kavin > OneDrive > Desktop > DAA > WEEK - 3 > C quicksort.c >
1 #include <stdio.h>
2
3 void swap(int arr[], int i, int j) {
4     int temp = arr[i];
5     arr[i] = arr[j];
6     arr[j] = temp;
7 }
8
9 int partition(int arr[], int low, int high) {
10    int p = arr[low];
11    int i = low;
12    int j = high;
13
14    while (i < j) {
15        while (arr[i] <= p && i <= high - 1)
16            i++;
17        while (arr[j] > p && j >= low + 1)
18            j--;
19        if (i < j)
20            swap(arr, i, j);
21    }
22    swap(arr, low, j);
23    return j;
24 }
25
26 void quickSort(int arr[], int low, int high) {
27    if (low < high) {
28        int pi = partition(arr, low, high);
29        quickSort(arr, low, pi - 1);
30        quickSort(arr, pi + 1, high);
31    }
32 }
33
34 int main() {
35    int n;
36    printf("Enter number of elements: ");
37    scanf("%d", &n);
38
39    int arr[n];
40
41    printf("Enter elements:\n");
42    for (int i = 0; i < n; i++)
43        scanf("%d", &arr[i]);
44
45    quickSort(arr, 0, n - 1);
46
47    printf("Sorted array:\n");
48    for (int i = 0; i < n; i++)
49        printf("%d ", arr[i]);
50
51    return 0;
52 }
```

```
C:\Users\kavin\OneDrive\Desktop\DA\WEEK - 3>gcc quicksort.c
```

```
C:\Users\kavin\OneDrive\Desktop\DA\WEEK - 3>a
Enter number of elements: 6
Enter elements:
12 31 3 12 5 6
Sorted array:
3 5 6 12 12 31
C:\Users\kavin\OneDrive\Desktop\DA\WEEK - 3>
```