

Week 5

30/05/2024

PAGE NO:
DATE:

Merge Sort :-

- Code :-

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

```
#include <time.h>
```

```
#include <stdlib.h>
```

```
void split(int[], int, int);
```

```
void combine(int[], int, int, int);
```

```
void main() {
```

```
    int a[5000], n, i, j, ch, temp;
```

```
    clock_t start, end;
```

```
    while (1) {
```

```
        printf("1: for main menu entry of N value & array elements");
```

```
        printf("2: To display time taken for N = 5000 to 10000");
```

```
        printf("10000 to 100000");
```

```
        printf("3: To exit ");
```

```
        printf("Enter your choice: ");
```

```
        scanf("%d", &ch);
```

```
        switch (ch) {
```

```
            case 1: printf("Enter the no of elms: ");
```

```
            scanf("%d", &n);
```

```
            printf("Enter array elements: ");
```

```
            for (i = 0; i < n; i++) {
```

```
                scanf("%d", &a[i]);
```

```
}
```

```
start = clock();
```

```
split(a, 0, n - 1);
```

```
end = clock();
```

```
printf("sorted array is: ");
```

```
for (i = 0; i < n; i++) {
```

```
    printf("%d ", a[i]);
```

```
    printf("Time taken to sort
```

```
q.d arr is %f secs",
```

```
n, ((double)(end - start)) /
```

```
(CLOCKS_PER_SEC);
```

```
break;
case 2: n = 5000;
while(n<=10500){
    for(i=0; i<n; i++)
        arr[i] = n - i;
    start = clock();
    split(a, 0, n-1);
    for(j=0; j<50000000; j++)
        temp = 38/600;
    end = clock();
    printf("Time taken to sort %d, %f\n",
           ((double)(end-start))/CLOCKS_PER_SEC);
    n = n + 1000;
}
break;
case 3: exit(0);
```

```
void split(int a[], int low, int high){
    int mid;
    if (low < high) {
        mid = (low + high) / 2;
        split(a, low, mid);
        split(a, mid + 1, high);
        combine(a, low, mid, high);
    }
}
```

void combine(int arr[], int low, int mid, int high) {

int c[15000], i, j, k;

i = low;

j = mid + 1;

while (i <= mid && j <= high) {

if (a[i] < a[j]) {

c[k] = a[i];

++k;

++i;

}

else {

c[k] = a[j];

++k;

++j;

}

}

if (i > mid) {

while (j <= high) {

c[k] = a[j];

++k;

++j;

}

}

if (j > high) {

while (i <= mid) {

c[k] = a[i];

++k;

++i;

}

}

for (i = low, j = high, i++) {

a[i] = c[i];

9

9

Output:-

1. For manual entry of N value & array elements.
2. To display time taken for sorting nnn N 500-14500
3. Exit

Enter your choice :-

Time taken to sort 500 numbers is 0.001417 secs

— u —	1500	— u —	0.004305 Secs
— u —	2500	— u —	0.009780 Secs
— u —	3500	— u —	0.0017931 Secs
— u —	4500	— u —	0.028694 Secs
— u —	5500	— u —	0.042466 Secs
— u —	6500	— u —	0.059055 Secs
— u —	7500	— u —	0.078023 Secs
— u —	8500	— u —	0.099680 Secs
— u —	9500	— u —	0.120517 Secs
— u —	10500	— u —	0.150043 Secs
— u —	11500	— u —	0.190248 Secs
— u —	12500	— u —	0.218345 Secs
— u —	13500	— u —	0.250040 Secs
— u —	14500	— u —	0.286645 Secs

AD
Soham

Algorithm :- combine($a[0 \dots n-1]$, low, mid, high)

// Input! $a[]$ is a sorted array from low to mid.

// ~~Output!~~ $a[]$ is a sorted array from mid+1 to high

// Output! An array $[0 \dots n-1]$ sorted in non-decreasing order.

$i \leftarrow low$

$j \leftarrow mid + 1$

$k \leftarrow low$

while $i <= mid$ and $j <= high$ do

if $a[i] < a[j]$

$c[k] \leftarrow a[i]$

$k \leftarrow k + 1$

$i \leftarrow i + 1$

else

$c[k] \leftarrow a[j]$

$k \leftarrow k + 1$

$j \leftarrow j + 1$

end if

end while

if $i > mid$ do

$c[k] \leftarrow a[i]$

$k \leftarrow k + 1$

$i \leftarrow i + 1$

end while

end if

for $i \leftarrow low$ to $high$ do

$a[i] \leftarrow c[i]$

end for.

Not by me
30/10/2019

Selection Sort & merge sort graph

N Value	SS Time (sec)	MS Time (sec)
500	0.003295	0.002698
1500	0.010922	0.002907
2500	0.01843	0.003065
3500	0.029585	0.003391
4500	0.040885	0.003037
5500	0.061843	0.002826
6500	0.063830	0.003703
7500	0.082681	0.003744
8500	0.105688	0.00321
9500	0.137809	0.003072
10500	0.161104	0.003149
11500	0.193703	0.003545
12500	0.22794	0.003321
13500	0.265655	0.003507
14500	0.305674	0.003624

