

29/05/24

Selection Sort

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

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

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

```
void selectsort (int a[], int n);
```

```
void main ()
```

```
{
```

```
int a[15000], n, i, j, ch, temp;
```

```
clock_t start, end;
```

```
while (1)
```

```
{
```

```
printf ("1: for manual entry N value & array  
elements");
```

```
printf ("2: To display time taken for sorting  
no. of elements N in range 500 to  
12,500");
```

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

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

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

```
switch (ch)
```

```
{
```

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

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

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

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

```
{
```

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

```
}
```



```

start = clock();
selectionsort(m, a);
end = clock();
printf("\n Sorted array is ");
for (i = 0; i < m; i++)
    printf("%d ", a[i]);
printf("\n Time taken sort %d no. %d sec", m,
    ((double)(end - start)) / clock() * 1000);
break;

```

case 2:

```

m = 500;
while (m <= 15000) {
    for (i = 0; i < m; i++)
        {
            a[i] = random(1000);
            a[i] = m - i;
        }
    start = clock();
    selectionsort(m, a);
    for (j = 0; j < 50000; j++)

```


29/05/24

Experiment 3

Merge sort

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

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

```
void main ()
{
```

```
    int a[15000], m, j, i, ch, temp;
    clock_t start, end;
```

```
    while (1)
    {
```

```
        printf ("Enter the no. of elements");
        scanf ("%d", &m);
        printf ("Enter array elements");
        for (i=0; i<m; i++)
        {
```

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

```
        start = clock();
        split (a, 0, m-1);
        end = clock();
```

```
        printf ("%d", a[i]);
        printf ("Time taken to sort %d no. of %d size array is: %f\n", m, a[i], ((double)(end-start))/CLOCKS_PER_SEC);
        break;
    }
```


case 2:

```

    n = 500;
    while (n <= 14500)
    {
        for (i = 0; i < n; i++)
        {
            a[i] = n - i;
        }
        start = clock();
        split(a, 0, n - 1);

        for (j = 0; j < 5000; j++) { tmp = 38 / 600; }
        end = clock();
        printf("Time taken to sort %d n is %f sec\n",
            ((double)(end - start)) // clocks for sec);
        n = n + 1000;
    }
    break;

```

case 3: print(0);

```

    {
        getchar();
    }

    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);
        }
    }

```


}

}

void condense (int a[], low, mid, high)

{

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

i = k = low;

j = mid + 1;

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

{

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

{

c[k] = a[i];

++k;

++i;

}

}

if (i > mid)

{

while (j <= high)

{

c[k] = a[j];

++k;

++j;

}

}

for (i = low; i <= high; i++)

{

a[i] = c[i];

}

}

Output:-

- 1) for manual entry of n values & array elements
- 2) To display time taken for sorting no. of elements
Var the range 500 to 14500
- 3) To exit

Enter the choice: 1

Enter the no. of elements: 4

Enter array elements: 44 77 22 11

Sorted array elements: 11 22 77 44

Time taken sort 4 no is 0.000012 sec

1) Enter the choice: 2

Time taken to sort 500 no is 0.002698 sec

1500 is 0.002907 sec

2500 is 0.003055 sec

3500 is 0.003391 sec

4500 is 0.003736 sec

5500 is 0.003702 sec

6500 is 0.003703 sec

7500 is 0.0037041 sec

8500 is 0.0041445 sec

9500 is 0.0051000 sec

graph:-

N Values	SS Time	Ms Time
5000	0.003295	0.002698
15000	0.010222	0.002907
25000	0.01843	0.003055
35000	0.02954	0.003037
45000	0.051843	0.003703
55000	0.0105688	0.003794
65000	0.0131809	0.003072
75000	0.0161104	0.003144
85000	0.193703	0.003546
95000	0.22794	0.003546
105000	0.265654	0.003507
115000	0.305674	0.003623
135000	0.306784	0.003728

Selection sort & merge sort

