

n: 6

CAB-05

classmate

Date

Page

30/05/2024

// Selection sort

// Algorithm :-

Sel-sort ( $a[0 \dots n-1]$ )

// sorts given array by selection sort.

// Input: An array  $a[0 \dots n-1]$  sorted in ascending order.

// output: Array  $a[0 \dots n-1]$  sorted in .

~~30/05/24~~



LAB - 05

// Selection Sort.

classmate

Date

Page

30/05/24

→ Algorithm:

Sel-sort( $a[0 \dots n-1]$ )  
// Sorts a given array by selection sort  
// Input: An array  $a[0 \dots n-1]$  of orderable elements.  
// Output: Array  $a[0 \dots n-1]$  sorted in ascending order

for  $i \leftarrow 0$  to  $n-2$  do

    Small-pos  $\leftarrow i$

        for  $j \leftarrow i+1$  to  $n-1$  do

            if  $a[j] < a[\text{Small-pos}]$

                Small-pos  $\leftarrow j$

        end if

    end for

        swap  $a[i]$  and  $a[\text{Small-pos}]$ .

end for.

→ program.

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

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

```
void main()
```

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

```
clock_t start, end;
```

```
while (1) {
```

```
    printf("\n 1: for manual entry of N value  
    and array elements");
```



classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```

printf("\n2: To display time taken for sorting
number of elements N in the range 500 to 14500");
printf("\n3: To exit");
printf("\n Enter your choice:");
scanf("%d", &ch);
switch (ch) {
    case 1:
        printf("\n Enter the number of elements:");
        scanf("%d", &n);
        printf("\n Enter the array elements: \n");
        for (i=0; i<n; i++) {
            scanf("%d", &a[i]);
        }
        start = clock();
        selectionSort(n, a);
        end = clock();
        printf("\n Sorted array is:");
        for (i=0; i<n; i++) {
            printf("%d ", a[i]);
        }
        printf("\n Time taken to sort %d numbers
is %.2f secs", n, ((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();
    selectionSort(n, a);

```



```
for (j = 0; j < 500000; j++) {
    temp = 38 / 600;
```

```
}
end = clock();
printf("Time taken to sort %d number  
is %.2f sec", n, ((double)(end - start)) /  
(CLOCKS_PER_SEC));
n = n + 1000;
```

```
}
break;
```

```
case 3:
    cout << endl;
```

```
}
getchar();
```

```
void selectionSort(int n, int a[]) {
    int i, j, t, small, pos;
    for (i = 0; i < n - 1; i++) {
        pos = i;
        small = a[i];
        for (j = i + 1; j < n; j++) {
            if (a[j] < small) {
                small = a[j];
                pos = j;
            }
        }
        t = a[i];
        a[i] = a[pos];
        a[pos] = t;
```

```
}
}
```



output:-

1. For Manual entry of N value & array elements
2. To display time taken for sorting number of elements N in the range 500 to 14500
3. exit

Enter your choice: 1

Enter the number of elements: 4

Enter array elements: 44 33 22 11

Sorted array is: 11 22 33 44

Time taken to sort 4 numbers is 0.000 sec

1. For Manual entry of N value & array elements
2. To display time taken for sorting number of elements N in the range 500 to 14500
3. exit

Enter your choice: 2

Time taken to sort 500 numbers is 0.000000 sec

Time taken to sort 1000 numbers is 0.000000 sec

Time taken to sort 2500 numbers is 0.000000 sec

Time taken to sort 3500 numbers is 0.060000 sec

Time taken to sort 4500 numbers is 0.010000 sec

Time taken to sort 5500 numbers is 0.016000 sec

Time taken to sort 6500 numbers is 0.015000 sec

Time taken to sort 7500 numbers is 0.016000 sec

Time taken to sort 8500 numbers is 0.016000 sec

Time taken to sort 9000 numbers is 0.032000 sec

Time taken to sort 10000 numbers is 0.047000 sec

Time taken to sort 11800 numbers is 0.047000 sec



Time taken to sort 12000 numbers is 0.046000 sec  
 Time taken to sort 13500 numbers is 0.063000 sec  
 Time taken to sort 14500 numbers is 0.062000 sec

1. for manual entry of no value and array elements
2. to display time taken for sorting number of elements N in the range 100 to 10000
3. to exit

Enter your choice: 3.



## // Merging

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

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

```
{
    int a[1000], n, i, j, ch, temp;
    clock_t start, end;
```

```
while (1)
```

```
{
```

```
    printf("Info: manual entry of N value & array elements");
```

```
    printf("Info: To display time taken for sorting number of elements N in range 100 to 1000");
```

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

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

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

```
    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();
```



classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```

split(a, 0, n-1);
end = clock();
printf("In sorted array is: ");
for (i = 0; i < n; i++)
    printf("%d ", a[i]);
printf("In time taken to sort %d numbers is  
%.4f secs", n, ((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();

```

for (j = 0; j < 50000; j++) temp = 38 / 600;
end = clock();

```

```

printf("In time taken to sort %d numbers  
is %.4f secs", n, ((double)(end - start)) /
(CLOCKS_PER_SEC));

```

n = n + 1000;

break;

case 3: exit(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);

combine(a, low, mid, high);

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

int c[1000], 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;

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; i <= high; i++)
    {
        a[i] = c[i];
    }
}

```

output:-

1. to manual entry of N value & array elements
  2. to display time taken for sorting no. of elements N in the range 500 to 148000
  3. exit.
- Enter your choice: )

Enter the no. of elements: 4

Enter array elements: 44 33 22 11



sorted array is: 11 22 33 44

Time taken to sort 24 numbers is 0.000000 sec

1. to manual entry of n value & array elements
2. To display time taken for sorting numbers in the range 800 to 14500
3. To exit

Enter your choice: 2

Time taken to sort	Numbers	Time taken to sort	Numbers	Time taken to sort	Numbers
0.000000	800	0.000000	8500	0.000000	14000
0.000000	8500	0.000000	9000	0.000000	14500
0.000000	9000	0.000000	9500	0.000000	15000
0.000000	9500	0.000000	10000	0.000000	15500
0.000000	10000	0.000000	10500	0.000000	16000
0.000000	10500	0.000000	11000	0.000000	16500
0.000000	11000	0.000000	11500	0.000000	17000
0.000000	11500	0.000000	12000	0.000000	17500
0.000000	12000	0.000000	12500	0.000000	18000
0.000000	12500	0.000000	13000	0.000000	18500
0.000000	13000	0.000000	13500	0.000000	19000
0.000000	13500	0.000000	14000	0.000000	19500
0.000000	14000	0.000000	14500	0.000000	20000

1. for manual entry of n value & array elements
2. To display time taken for sorting numbers in the range 800 to 14000
3. To exit

Enter your choice: 3