30/May/24

Lab-5

Selection Sort for large inputs.

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

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

int main () {
    int a[15000], n, i, j, ch, temp;
    clock_t start, end;

    while (1) {

Printf ("\n1: for manual Entry of N value
                and array elements");

Printf ("\n2: To display time taken for
            Sorting number of elements N
        in the range 500 to 14500");

Printf ("\n3: To Exit");
Printf ("\nEnter Your Choice:");
    Scanf ("%d", &ch);

    Switch (ch) {
    case 1:
        Printf ("\nEnter the number of
                    Elements:");
        Scanf ("%d", &n);
```

```
Printf ("Enter array elements:");
    for (i=0 ; i<n ; i++) {
        Scanf ("%d", &a[i]);
    }
    start = clock();
    Selsort (n, a);
    end = clock();
    Printf ("\n Sorted array is :");
    for (i=0; i<n; i++) {
        Printf ("%d\t", a[i]);
    }
Printf ("\n Time taken to sort %d numbers
    is %f 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();
    Selsort (n, a);
    for (j=0; j < 500000; j++) {
        temp = 38/600;
```

```
    }
    end = clock();
    Printf("\n Time taken to sort %d
numbers is %f secs", n,
        ((double).(end - start)) / Clock_PER
    n+= 1000;
    }
    break;

case 3:
    exit(0);
}
getchar();
}
return 0;
}

void selsort (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 and array elements

2: Time taken for sorting number of elements N in the range 500 to 14500

3: To 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

Enter your choice: 2

Time Taken to Sort 500 number is 0.000000 secs

Time Taken to sort 1500 number is 0.000000 secs

Time Taken to Sort 2500 number is 0.000000 secs

Time Taken to sort 3500 numbers is 0.015000 secs

Time taken to Sort 4500 numbers is 0.0000000 Secs

Time taken to Sort 5500 Numbers is 0.0160000 Secs

Time taken to Sort 6500 numbers is 0.015000 Secs

Time taken to Sort 7500 numbers is 0.016000 Secs

Time taken to Sort 8500 Numbers is 0.016000 Secs

Time taken to Sort 9500 Numbers is 0.031000 Secs

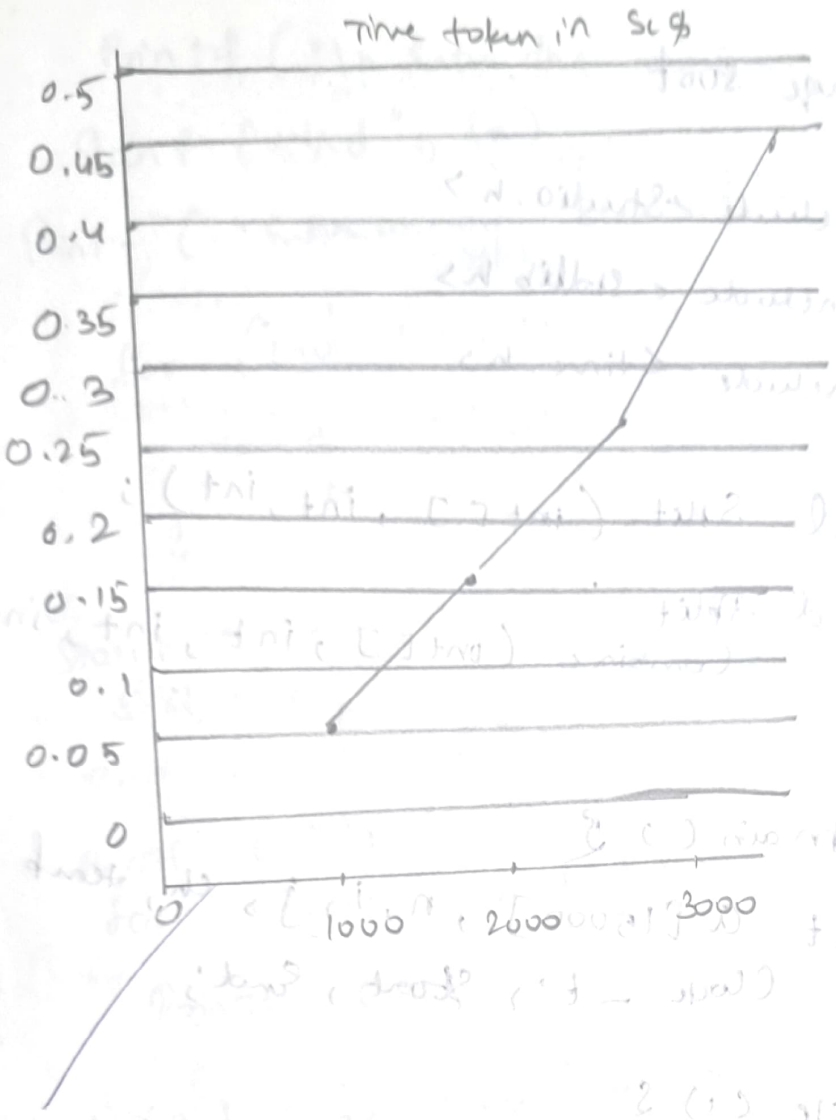
Time taken to Sort 10500 number is 0.047000 Secs

Time taken to Sort 11500 number is 0.047000 Secs

Time taken to Sort 12500 number is 0.047000 Secs

Time taken to Sort 13500 number is 0.046000 Secs

Time taken to Sort 14500 Number is 0.63000 Secs


Time token in Sec
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000

2) Merge Sort

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

void Sort (int[], int, int);
void Combine (int[], int, int, int);

int main() {
    int a[15000], n, i, j, ch, temp;
    clock_t, start, end;

    while (1) {
Printf ("\n1: for manual entry of N value
        and array elements");
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 ("Enter array Elements:");
for (i=0; i<n; i++){
    Scanf ("%d", &a[i]);
}
start = clock();
split (a, 0, n-1);
end = clock();
Printf ("\n Sorted array is:");
for (i=0; i<n; i++){
    Printf ("%d\t", a[i]);
Printf ("\n Time taken to Sort %d numbers is
    %f secs", n, ((double) (end-start)
    / CLOCKS_PER_SEC));
n = n+1000;
}
break;
Case 3:
    Exit (0);
}
getchar();
}
return 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 a[], int low, int mid,
              int 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;
    }
    else {
      c[k] = a[j];
      ++k;
      ++j;
    }
  }
  if (i > mid) {
    while (j <= high) {
      c[k] = a[j];
      ++k;
      ++j;
    }
  }
}
```

```
for (i=low; i<=high ;i++) {
    a[i] = c[i];
}
}
```

Output:

1) form manual Entry of N value and array Element

2. To display time taken for Sorting number of Elements N in the range 500 to 14500

3: To Exit

Enter your choice: 1

Enter the number of Element 4

Enter array Elements : 44 33 22 11

Sorted array is 11 22 33 44

Time taken to sort 4 number is 0.0000012 sec

~~2. To~~

Enter your choice: 2

Time taken to sort 500 number is 0.002698 sec

Time taken to sort 1500 number is 0.002907 sec

Time taken to sort 2500 number is 0.003291 sec

Time taken to Sort 3500 number is 0.003037 sec

Time taken to sort 4600 number is 0.002826 sec

Time taken to Sort 5500 number is 0.003703 sec

Time taken to Sort 6500 number is 0.003021 sec

Time taken to sort 7500 number is 0.003744

Time taken to Sort 8500 number is 0.003072 Secs

Time taken to sort 9500 number is 0.003144 Secs

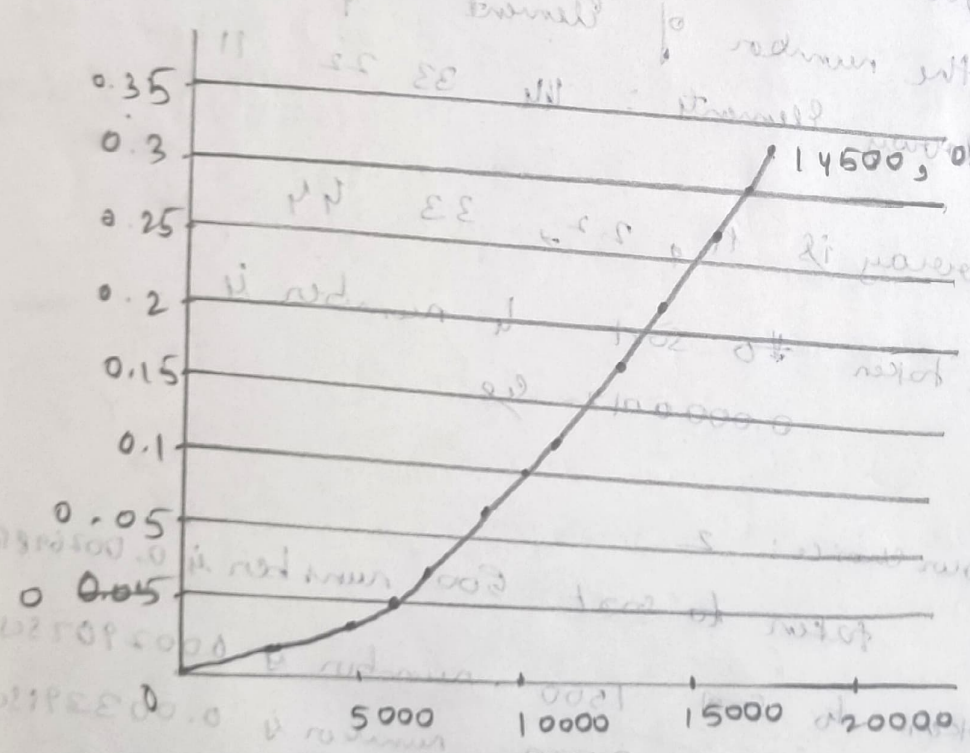
Time taken to sort 10500 number is 0.003546 Secs

Time taken to Sort 11500 number is 0.003321 Secs

Time taken to Sort 12500 number is 0.000324 Secs

Time taken to Sort 13500 number is 0.003507 Secs

Time taken to sort 14500 number is 0.003624 Secs



3/5/24