

6/6/24

LAB-5 Q4

```
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
int partition(int arr[], int low, int high)
```

pivot = arr[high]

int i = low - 1;

```
for (int j = low; j <= high; j++) {
```

if (arr[j] < pivot) {

i++;

```
swap(&arr[i], &arr[j]);
```

```
swap(&arr[i+1], &arr[high]);
```

```
return (i+1);
```

```
void quicksort(int arr[], int low, int high)
```

{

if (low < high)

```
int pi = partition(arr, low, high)
```

```
quicksort(arr, low, pi-1);
```

```
quicksort(arr, pi+1, high);
```

}

```

void main () {
    int a [5000], n, i, j, ch, temp;
    clock_t start, end;

    while (1) {
        if ("In 1: Enter N value & elements");
        if ("In 2: To display time taken for sorting
            number of elements N in range 500 to
            140000");
        if ("In 3: To exit");
        if ("In Enter choice:");
        scanf ("%d", &ch);

        switch (ch) {
            case 1:
                if ("h Enter the no. of elements");
                scanf ("%d", &n);
                printf ("Enter elements");
                for (i=0; i<n; i++) {
                    scanf ("%d", &a[i]);
                }

                start = clock();
                quicksort (a, 0, n-1);
                end = clock();

                if ("h Is sorted array:");
                for (i=0; i<n; i++) {
                    if ("%d (%f, a[i])");
                }

                if ("h Time taken to sort %d numbers is
                    %.f sec", n, ((double)(end - start)) /
                    (clocks_per_sec));
            break;
        }
    }
}

```

Case 2:

n=5000;

while (n <= 14500) {

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

a[i] = n-i;

} start = clock();

quicksort (a, 0, n-1);

for (j=0; j<5000000; j++) {

} temp = 0.38 / 600;

} end = clock();

printf (" \n Time taken to sort %d
numbers is %.f secs ", n,
(((double) (end - start)) / clock().cnsec));

n=n+1000;

}

break;

Case 3:

exit (0);

}

getchar();

,

.

P.6(4)24

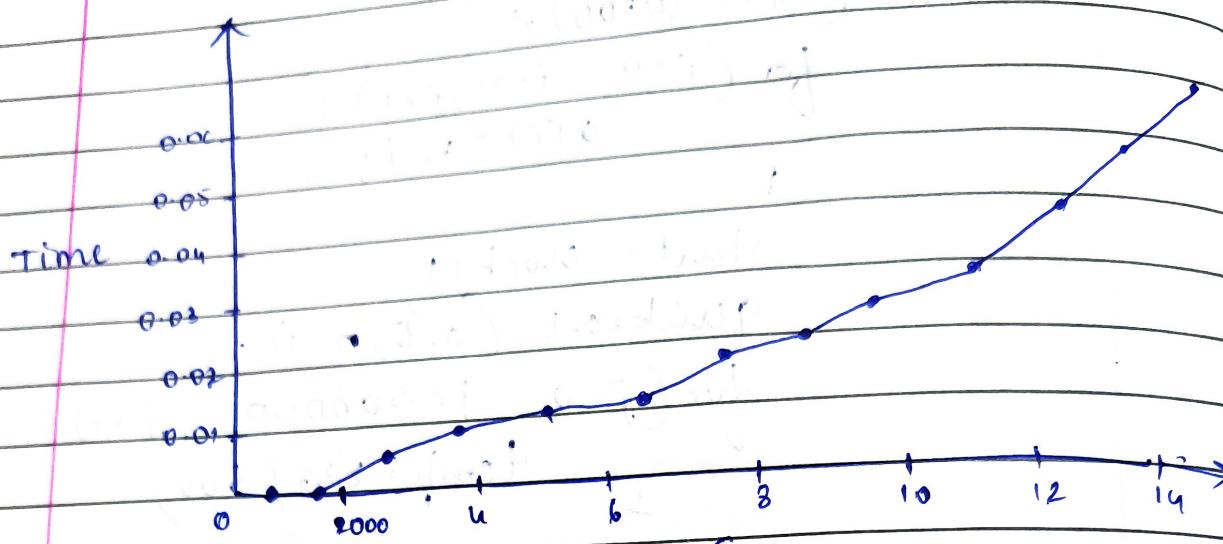
Output:

Enter no. of elements: 6

Enter array elements: 38 4 5 6 20 50

Sorted array is: 4 5 6 20 38 50

Graph:

 $N \rightarrow (k)$