**ADA LAB-9**

* **QUICK SORT WITH TIME COMPLEXITY**
  + **PROGRAM**

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

void swap(int \*a, int \*b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

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

{

int pivot = arr[high];

int i = (low - 1);

for (int j = low; j <= high - 1; 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 arr[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("\nEnter array elements: ");

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

{

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

}

start = clock();

quickSort(arr, 0, n - 1);

end = clock();

printf("\nSorted array is: ");

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

printf("%d\t", arr[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++)

{

arr[i] = n - i;

}

start = clock();

quickSort(arr, 0, n - 1);

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)) / CLOCKS\_PER\_SEC));

n = n + 1000;

}

break;

case 3:

exit(0);

}

getchar();

}

}

* + **OUTPUT**



