Practical 2

19BCE248

AIM: To implement Quick Sort algorithm by randomly selecting any element of an array as the pivot element. Display the output after each call to the "PARTITION" function finishes

Code:

#include <stdio.h>

#include <stdlib.h>

int \*a;

void quick\_sort(int p, int q,int n);

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

{

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}

int partion( int p, int r)

{

int pivotIndex = p + rand()%(r - p + 1);

int pivot;

int i = p - 1;

int j;

pivot = a[pivotIndex];

swap(&a[pivotIndex], &a[r]);

for (j = p; j < r; j++)

{

if (a[j] < pivot)

{

i++;

swap(&a[i], &a[j]);

}

}

swap(&a[i+1], &a[r]);

return i + 1;

}

void quick\_sort( int p, int q,int n)

{

int j;

if (p < q)

{

j = partion( p, q);

printf("Pivot Element:%d\n",a[j]);

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

printf("%d ",a[i]);

}

printf("\n");

quick\_sort( p, j-1,n);

quick\_sort( j+1, q,n);

}

}

int main()

{

int n;

scanf("%d",&n);

// int a[n];

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

// a[i]=rand();

// }

// int n=10;

a=(int\*)malloc(sizeof(int)\*n);

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

a[i]=rand();

}

quick\_sort( 0, n-1,n);

printf("\nFinal Sorted Array:\n");

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

printf("%d ", a[i]);

return 0;

}

Output:

