AIM:-

Aassignment 5

Implement Quick Sort to sort the given list of numbers. Display corresponding list in each pass.

OBJECTIVE:-

To study the quickest sorting technique .To know it’s implementation .

Theory :- QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.

1 .Always pick first element as pivot.

2 .Always pick last element as pivot (implemented below)

3 .Pick a random element as pivot.

4 .Pick median as pivot.

The key process in quickSort is partition(). Target of partitions is, given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x) before x, and put all greater elements (greater than x) after x. All this should be done in linear time.

Sourcecode :-

#include<iostream>

using namespace std;

int partion(int a[],int p,int r)

{

int x,i,j,l,temp,temp1;

x=a[r];

i=p-1;

for(j=p;j<=(r-1);j++)

{

if(a[j]<x)

{

i=i+1;

temp=a[i];

a[i]=a[j];

a[j]=temp;

cout<<a[i]<<" "<<a[j];

}

}

temp1=a[i+1];

a[i+1]=a[r];

a[r]=temp1;

return (i+1);

}

int qsort(int a[],int p,int r)

{

int q,l;

if(p<r)

{

q=partion(a,p,r);

qsort(a,p,q-1);

qsort(a,q+1,r);

}

cout<<endl;

for(l=0;l<(r+1);l++)

{

cout<<a[l]<<" ";

}

}

int main()

{

int a[100],i,j,n;

cout<<"How many number you have :";

cin>>n;

cout<<"Enter the element :"<<endl;

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

{

cin>>a[i];

}

cout<<endl<<"Your list is :";

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

{

cout<<a[i]<<" ";

}

cout<<endl;

qsort(a,0,(n-1));

return 0;

}

Output:-

/\*

How many number you have :4

Enter the element :

5

12

1

7

Your list is :5 12 1 7

5 51 12

1 5

1 5

1 5 7 12

1 5 7 12

--------------------------------

Process exited after 19.45 seconds with return value 0

Press any key to continue . . .

\*/

Conclusion:-

1. It is the quickest sorting algorithm .
2. With time complexity is O(nlog(n)).