**Experiment 8**

**Aim** : To write a program to implement quick sort

**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 quick sort that pick pivot in different ways. The key process in quick sort 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.

**Algorithm :**

1. Start
2. Take the first element as the smallest element and the last element as the largest
3. Take the last element as pivot, place it in the correct position in the sorted array
4. Create a recursive function quicksort() to place all elements than pivot to the left of the pivot and all elements larger than pivot to the right of pivot
5. Create a function to print the sorted array
6. In main() function get the array and size of array and pass it to quicksort()
7. Stop.

**Program :**

#include <iostream>

using namespace std;

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

{

int t = \*a;

\*a = \*b;

\*b = t;

}

int qpartition (int a[], int small, int large)

{

int pivot = a[large];

int i = (small - 1);

for (int j = small; j <= large- 1; j++)

{

if (a[j] <= pivot)

{

i++;

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

}

}

aswap(&a[i + 1], &a[large]);

return (i + 1);

}

void quickSort(int a[], int small, int large)

{

if (small < large)

{

int pi = qpartition(a, small, large);

quickSort(a, small, pi - 1);

quickSort(a, pi + 1, large);

}

}

void disp(int a[], int s)

{

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

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

cout<<"\n";

}

int main()

{

int n,i;

cout<<"Enter the no. of Elements for array :- ";

cin>>n;

int arr[n];

cout<<"Enter the array :- ";

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

cin>>arr[i];

cout<<"\n";

quickSort(arr, 0, n-1);

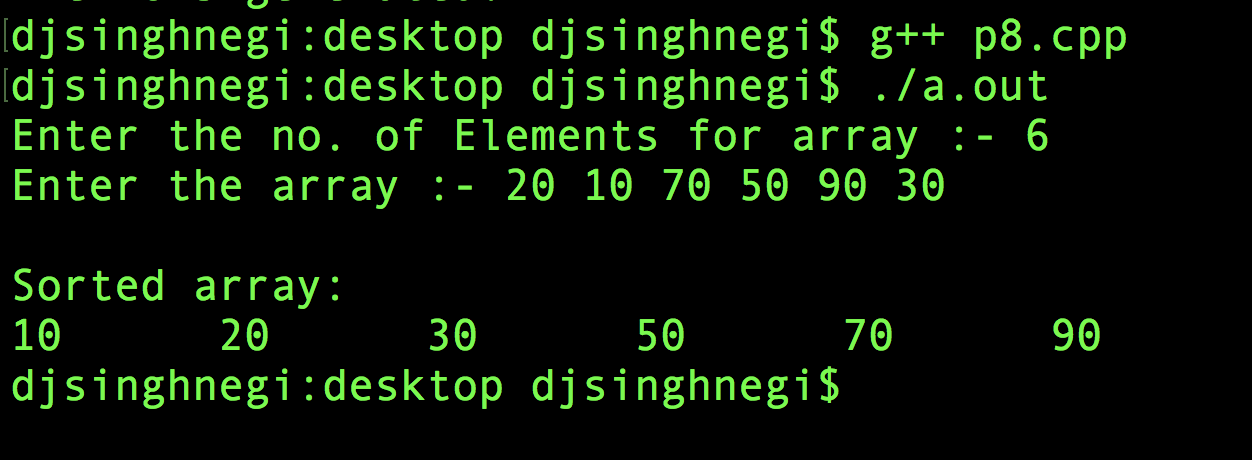
cout<<"Sorted array: \n";

disp(arr, n);

return 0;

}

**Output :**



**Learning :**

Quick Sort is an efficient implementation of sorting of array