Sorting Algorithms Set 2 **(14/Apr/2020)**

1. Quick Sort

**Source Code :**

#include <iostream>

using namespace std;

void quick\_sort(int[], int, int);

int partition(int[], int, int);

int main()

{

int a[50], n, i;

cout<<"Enter number of elements";

cin>>n;

cout<<"\nEnter array elements:";

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

cin>>a[i];

}

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

cout<<"\nArray after sorting:";

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

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

}

return 0;

}

void quick\_sort(int a[], int l, int u)

{

int j;

if(l < u){

j = partition(a, l, u);

quick\_sort(a, l, j - 1);

quick\_sort(a, j + 1, u);

}

}

int partition(int a[], int l, int u)

{

int v, i, j, temp;

v = a[l];

i = l;

j = u + 1;

do{

do{

i++;

}while(a[i] < v && i <= u);

do{

j--;

}while(v < a[j]);

if(i < j){

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}while(i < j);

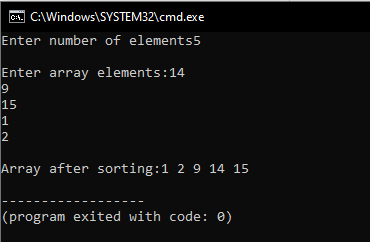
a[l] = a[j];

a[j] = v;

return(j);

}

**OUTPUT**



1. Merge sort

**Source Code :**

#include<stdio.h>

void mergesort(int a[], int i, int j);

void merge(int a[], int i1, int j1, int i2, int j2);

int main(){

int a[30], n, i;

printf("Enter no of elements:");

scanf("%d", &n);

printf("Enter array elements:");

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

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

}

mergesort(a, 0, n - 1);

printf("\nSorted array :");

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

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

}

return 0;

}

void mergesort(int a[], int i, int j)

{

int mid;

if(i < j){

mid = (i + j) / 2;

mergesort(a, i, mid);

mergesort(a, mid + 1, j);

merge(a, i, mid, mid + 1, j);

}

}

void merge(int a[], int i1, int j1, int i2, int j2)

{

int temp[50];

int i, j, k;

i = i1;

j = i2;

k = 0;

while(i <= j1 && j <= j2){

if(a[i] < a[j]){

temp[k++] = a[i++];

}

else{

temp[k++] = a[j++];

}

}

while(i <= j1){

temp[k++] = a[i++];

}

while(j <= j2){

temp[k++]=a[j++];

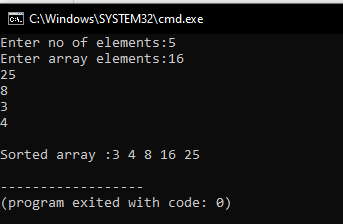
}

for(i = i1, j = 0; i <= j2; i++, j++)

a[i] = temp[j];

}

**OUTPUT**



<https://github.com/harinarayanank/Competitive-Lab/tree/master/Sorting%20Algorithms%20Set%202>