**NAME : GOKUL KONJETI**

**REG NO: 192111038**

**SUB: DATA STRUCTURS**

**SUB CODE: CSA0370**

16.INSERTION SORT

#include<stdio.h>

int insertion\_sort(int A[],int n)

{

int i,j,temp;

for(i=1;i<n;i++)

{

j=i;

while(j>0 && A[j]<A[j-1])

{

temp = A[j-1];

A[j-1]=A[j];

A[j]=temp;

j--;

}

}

}

int main()

{

int i,n,A[100];

printf("\nEnter n");

scanf("%d",&n);

printf("\nEnter Array elements");

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

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

insertion\_sort(A,n);

printf("\nAfter Sorting");

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

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

return 0;

}

16.QUICK SORT

#include<stdio.h>

void quicksort(int number[25],int first,int last){

int i, j, pivot, temp;

if(first<last){

pivot=first;

i=first;

j=last;

while(i<j){

while(number[i]<=number[pivot]&&i<last)

i++;

while(number[j]>number[pivot])

j--;

if(i<j){

temp=number[i];

number[i]=number[j];

number[j]=temp;

}

}

temp=number[pivot];

number[pivot]=number[j];

number[j]=temp;

quicksort(number,first,j-1);

quicksort(number,j+1,last);

}

}

int main(){

int i, count, number[25];

printf("How many elements are u going to enter?: ");

scanf("%d",&count);

printf("Enter %d elements: ", count);

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

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

quicksort(number,0,count-1);

printf("Order of Sorted elements: ");

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

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

return 0;

}

17.MERGE SORT

#include <stdio.h>

void merge\_Sort(int [], int, int, int);

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

main()

{

int a[50] , i, n;

printf("Enter total number of elements:");

scanf("%d", &n);

printf("Enter the elements:\n");

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

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

partition( a, 0, n - 1);

printf("After merge sort:\n");

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

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

}

void partition(int a[],int low,int high)

{

int mid;

if(low < high)

{

mid = (low + high)/2;

partition( a, low, mid);

partition(a, mid+1, high);

merge\_Sort(a, low, mid, high);

}

}

void merge\_Sort(int a[], int low, int mid, int high)

{

int i, j, k, lo, temp[50];

lo = low;

i = low;

j = mid + 1;

while ((lo <= mid) && (j <= high))

{

if (a[lo] <= a[j])

{

temp[i] = a[lo];

lo++;

}

else

{

temp[i] = a[j];

j++;

}

i++;

}

if (lo > mid)

{

for (k = j; k <= high; k++)

{

temp[i] = a[k];

i++;

}

}

else

{

for (k = lo; k <= mid; k++)

{

temp[i] = a[k];

i++;

}

}

for (k = low; k <= high; k++)

a[k] = temp[k];

}

19.HEAP SORT

#include<stdio.h>

heapify(int a[], int n, int i)

{

int root,l,r,t;

root = i;

l = 2\*i + 1;

r = 2\*i + 2;

if (l<n && a[l] > a[root])

root = l;

if (r<n && a[r] > a[root])

root = r;

if (root != i)

{

t=a[i];

a[i] = a[root];

a[root] = t;

heapify(a, n, root);

}

}

heapsort(int a[], int n)

{

int i,t;

for(i=n/2-1; i>=0; i--)

heapify(a, n, i);

for(i=n-1; i>=0; i--)

{

t= a[0];

a[0]= a[i];

a[i] = t;

heapify(a, i, 0);

}

}

int main()

{

int a[50],i,n;

printf("Enter total number of elements:");

scanf("%d", &n);

printf("Enter the elements:\n");

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

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

heapsort(a,n);

printf("\n\nAfter Heap sort:\n");

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

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

return 0;

}