heapsort

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Faculty

{

int id;

int subject\_code;

char name[100];

char class[20];

};

int z = 0;

void swap(struct Faculty \* f ,int first,int last){

int temp;

char tempchr[100];

temp = f[first].subject\_code;

f[first].subject\_code = f[last].subject\_code;

f[last].subject\_code = temp;

temp = f[first].id;

f[first].id = f[last].id;

f[last].id = temp;

strcpy(tempchr,f[first].name);

strcpy(f[first].name,f[last].name);

strcpy(f[last].name,tempchr);

strcpy(tempchr,f[first].class);

strcpy(f[first].class,f[last].class);

strcpy(f[last].class,tempchr);

z++;

}

void heapify(struct Faculty \* arr, int n, int i) {

int max = i;

int leftChild = 2 \* i + 1;

int rightChild = 2 \* i + 2;

if (leftChild < n && arr[leftChild].id > arr[max].id)

max = leftChild;

if (rightChild < n && arr[rightChild].id > arr[max].id)

max = rightChild;

if (max != i) {

swap(arr,i,max);

heapify(arr, n, max);

}

}

void heapSort(struct Faculty \* arr, int n) {

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

heapify(arr, n, i);

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

swap(arr,0,i);

heapify(arr, i, 0);

}

}

void display(struct Faculty \* arr, int n) {

int i;

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

printf("Faculty name: %s\n", arr[i].name);

printf("Faculty class name: %s\n", arr[i].class);

printf("Facultys id: %d\n", arr[i].id);

printf("Facultys subject code: %d\n", arr[i].subject\_code);

}

}

int main() {

int i, n;

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

scanf("%d",&n);

struct Faculty arr[n];

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

printf("Enter %d Facultys id: ", i+1);

scanf("%d",&arr[i].id);

printf("Enter %d Faculty name: ", i+1);

scanf("%s",&arr[i].name);

printf("Enter %d Faculty class: ", i+1);

scanf("%s",&arr[i].class);

printf("Enter %d Facultys subjectcode: ", i+1);

scanf("%d",&arr[i].subject\_code);

}

printf("\n Non-sorted array:\n");

display(arr, n);

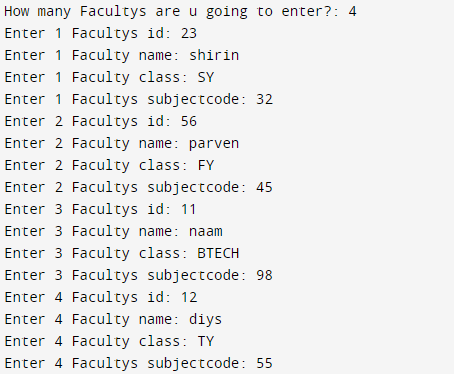
heapSort(arr, n);

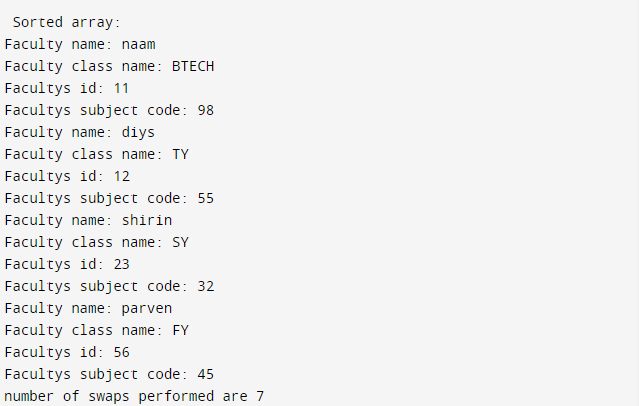
printf("\n Sorted array:\n");

display(arr, n);

printf("number of swaps performed are %d\n",z);

}





Merge Sort

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Faculty

{

int ID;

int subject\_code;

char name[100];

char class[100];

};

int x = 0 ;

void assign(struct Faculty \* f ,int first,struct Faculty \* g,int last){

f[first].subject\_code = g[last].subject\_code;

f[first].ID = g[last].ID;

strcpy(f[first].name,g[last].name);

strcpy(f[first].class,g[last].class);

}

void merge(struct Faculty \* arr, int start, int mid, int end) {

int len1 = mid - start + 1;

int len2 = end - mid;

struct Faculty leftArr[len1], rightArr[len2];

for (int i = 0; i < len1; i++){

assign(leftArr, i, arr,start + i);

x++;

}

for (int j = 0; j < len2; j++){

assign(rightArr,j ,arr,mid + 1 + j);

x++;

}

int i, j, k;

i = 0;

j = 0;

k = start;

while (i < len1 && j < len2) {

if (leftArr[i].ID <= rightArr[j].ID) {

assign(arr,k,leftArr,i);

i++;

x++;

} else {

assign(arr,k,rightArr,j);

j++;

x++;

}

k++;

}

while (i < len1) {

assign (arr,k,leftArr,i);

i++;

k++;

x++;

}

while (j < len2) {

assign(arr,k,rightArr,j);

j++;

k++;

x++;

}

}

void mergeSort(struct Faculty \* arr, int start, int end) {

if (start < end) {

int mid = start + (end - start) / 2;

mergeSort(arr, start, mid);

mergeSort(arr, mid + 1, end);

merge(arr, start, mid, end);

}

}

void display(struct Faculty \* arr, int sixe) {

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

printf("Faculty name: %s\n", arr[i].name);

printf("Faculty class name: %s\n", arr[i].class);

printf(" Facultys id: %d\n", arr[i].ID);

printf("Facultys subject code: %d\n", arr[i].subject\_code);

}

printf("\n");

}

int main() {

int i, count;

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

scanf("%d",&count);

struct Faculty arr[count];

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

printf("Enter Faculty ID: ");

scanf("%d",&arr[i].ID);

printf("Enter Faculty name: ");

scanf("%s",&arr[i].name);

printf("Enter Faculty's class name: ");

scanf("%s",&arr[i].class);

printf("Enter Facultys subject code:");

scanf("%d",&arr[i].subject\_code);

}

printf("Original array\n");

display(arr, count);

mergeSort(arr, 0, count - 1);

printf("Sorted array\n");

display(arr, count);

printf("Number of comparisions are %d\n",x);

}

