DS Lab Assignment 1

Pranav Joshi
CS-B
Roll no: 43

Title: WAP to implement Quick sort, Merge Sort and Heap Sort on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no. And count the number of swap performed.

Code:

1. Quick Sort

```c
#include<stdio.h>

#include <stdlib.h>
#include <string.h>
struct employee
{
    int emp_no;
    int emp_salary;
    char emp_name[100];
};
int n = 0 ;
void swap(struct employee * f ,int first,int last){
    int temp;
    char tempchr[100];
    temp = f[first].emp_salary;
    f[first].emp_salary = f[last].emp_salary;
    f[last].emp_salary = temp;
    temp = f[first].emp_no;
    f[first].emp_no = f[last].emp_no;
    f[last].emp_no = temp;
    strcpy(tempchr,f[first].emp_name);
    strcpy(f[first].emp_name,f[last].emp_name);
    strcpy(f[last].emp_name,tempchr);
}
void quicksort(struct employee * f ,int first,int last){
   int i, j, pivot, temp;
   if(first<last){
      pivot=f[first].emp_no;
      i=first;
      j=last;
      while(i<j){
         while(f[i].emp_no<=pivot&&i<last){
             i++;
        }
```

```
        while(f[j].emp_no>pivot){
            j--;
        }
        if(i<j){
            swap(f,i,j);
            n++;
        }
    }
    swap(f,j,first);
    n++;
    quicksort(f,first,j-1);
    quicksort(f,j+1,last);
  }
}
int main(){
  int i, count;
  printf("Enter number of employees: ");
  scanf("%d",&count);
  struct employee arr[count];
  for(i=0;i<count;i++){
      printf("Enter %d emp_no: ", i+1);
      scanf("%d",&arr[i].emp_no);
      printf("Enter %d emp_name: ", i+1);
      scanf("%s",&arr[i].emp_name);
      printf("Enter %d emp_salary: ", i+1);
      scanf("%d",&arr[i].emp_salary);
   }
  quicksort(arr,0,count-1);
  printf("Order of Sorted elements: \n");
  for(i=0;i<count;i++){
      printf("emp_name: %s\n", arr[i].emp_name);
      printf("emp_no: %d\n", arr[i].emp_no);
      printf("emp_salary: %d\n", arr[i].emp_salary);
  }
  printf("number of swaps performed are %d\n",n);
  return 0;
}
```

Output given below:

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Code\C\Code> cd "c:\Code\C\Code\" ; if ($?) { gcc quicksort.c -o quicksort } ; if ($?) { .\quicksort }
Enter number of employees: 3
Enter 1 emp_no: 9
Enter 1 emp_name: aabb
Enter 1 emp_salary: 4000
Enter 2 emp_no: 5
Enter 2 emp_name: ccdd
Enter 2 emp_salary: 5000
Enter 3 emp_no: 7
Enter 3 emp_name: eeff
Enter 3 emp_salary: 3000
Order of Sorted elements:
emp_name: ccdd
emp_no: 5
emp_salary: 5000
emp_name: eeff
emp_no: 7
emp_salary: 3000
emp_name: aabb
emp_no: 9
emp_salary: 4000
number of swaps performed are 2
PS C:\Code\C\Code>
```

2. Merge Sort

```c
#include <stdio.h>

#include <stdlib.h>
#include <string.h>
struct employee
{
    int emp_no;
    int emp_salary;
    char emp_name[100];
};

void assign(struct employee * f ,int first,struct employee * g,int last){
    f[first].emp_salary = g[last].emp_salary;
    f[first].emp_no = g[last].emp_no;
    strcpy(f[first].emp_name,g[last].emp_name);
}

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

  int len1 = mid - start + 1;
  int len2 = end - mid;

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

  for (int i = 0; i < len1; i++){
    assign(leftArr, i, arr,start + i);
```

```
  }
  for (int j = 0; j < len2; j++){
    assign(rightArr,j ,arr,mid + 1 + j);

  }
  int i, j, k;
  i = 0;
  j = 0;
  k = start;

  while (i < len1 && j < len2) {
    if (leftArr[i].emp_no <= rightArr[j].emp_no) {
      assign(arr,k,leftArr,i);
      i++;

    } else {
      assign(arr,k,rightArr,j);
      j++;
    }
    k++;
  }

  while (i < len1) {
    assign (arr,k,leftArr,i);
    i++;
    k++;
  }

  while (j < len2) {
    assign(arr,k,rightArr,j);
    j++;
    k++;
  }
}

void mergeSort(struct employee * 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 employee * arr, int size) {
  for(int i=0;i<size;i++){
       printf("emp_name: %s\n", arr[i].emp_name);
       printf("emp_no: %d\n", arr[i].emp_no);
```

```
        printf("emp_salary: %d\n", arr[i].emp_salary);
    }
   printf("\n");
}
int main() {
    int i, count;
    printf("Enter the number of employees: ");
    scanf("%d",&count);
    struct employee arr[count];
   for(i=0;i<count;i++){
        printf("Enter emp_no: ");
        scanf("%d",&arr[i].emp_no);
        printf("Enter emp_name: ");
        scanf("%s",&arr[i].emp_name);
        printf("Enter emp_salary: ");
        scanf("%d",&arr[i].emp_salary);
     }
   printf("Original array\n");
   display(arr, count);
   mergeSort(arr, 0, count - 1);
   printf("Sorted array\n");
   display(arr, count);
}
```

Output:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\Code\C\Code> cd "c:\Code\C\Code\" ; if ($?) { gcc mergesort.c -o mergesort } ; if ($?) { .\mergesort }
Enter the number of employees: 3
Enter emp_no: 9
Enter emp_name: aabb
Enter emp_salary: 4000
Enter emp_no: 5
Enter emp_name: ccdd
Enter emp_salary: 5000
Enter emp_no: 7
Enter emp_name: eeff
Enter emp_salary: 3000
Original array
emp_name: aabb
emp_no: 9
emp_salary: 4000
emp_name: ccdd
emp_no: 5
emp_salary: 5000
emp_name: eeff
emp_no: 7
emp_salary: 3000

Sorted array
emp_name: ccdd
emp_no: 5
emp_salary: 5000
emp_name: eeff
emp_no: 7
emp_salary: 3000
emp_name: aabb
emp_no: 9
emp_salary: 4000

PS C:\Code\C\Code>
```

Merge sort does not need or have swaps. Therefore, number of swaps performed=0.

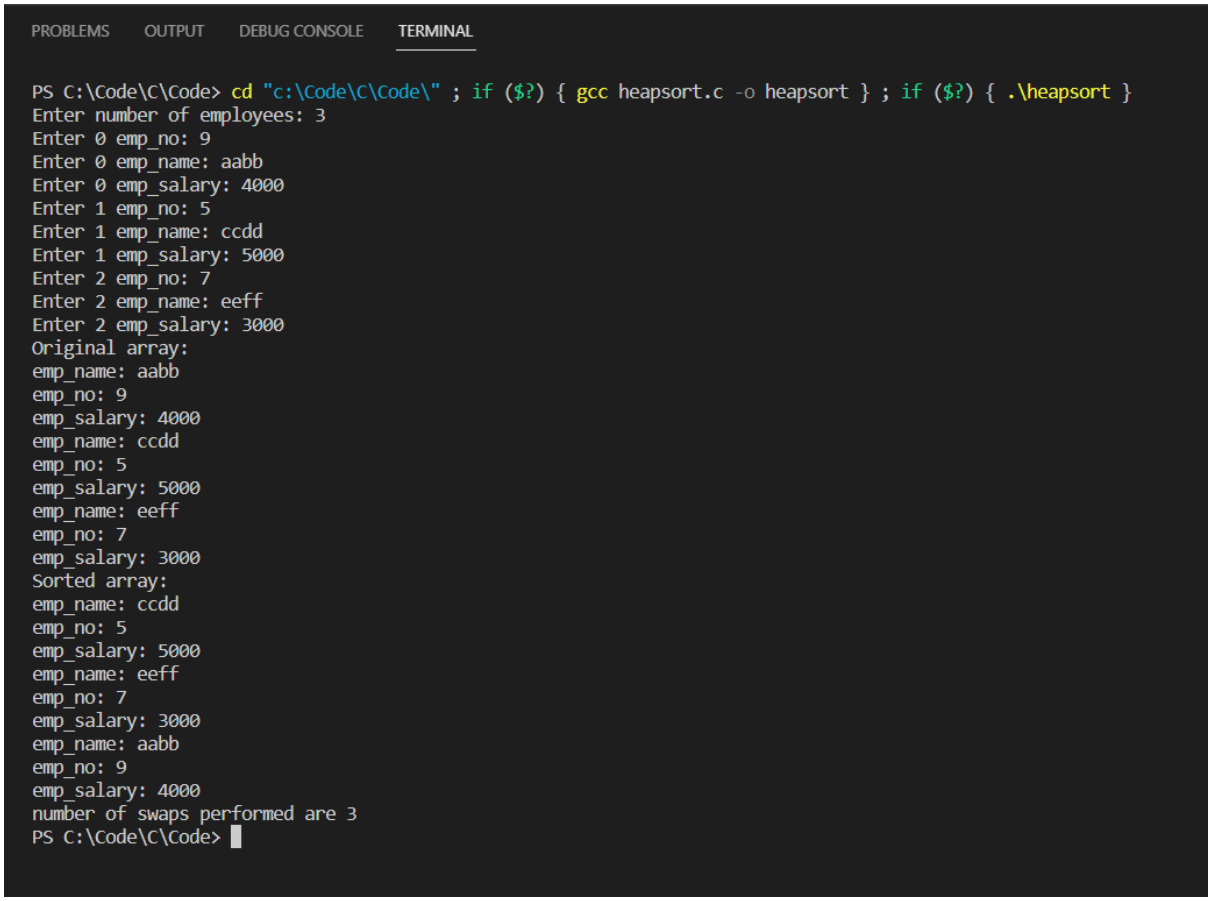
3. Heap Sort

```c
#include <stdio.h>

#include <stdlib.h>
#include <string.h>
struct employee
{
  int emp_no;
  int emp_salary;
  char emp_name[100];
};
int z = 0;
void swap(struct employee * f ,int first,int last){
    int temp;
    char tempchr[100];
    temp = f[first].emp_salary;
    f[first].emp_salary = f[last].emp_salary;
    f[last].emp_salary = temp;
    temp = f[first].emp_no;
    f[first].emp_no = f[last].emp_no;
    f[last].emp_no = temp;
    strcpy(tempchr,f[first].emp_name);
    strcpy(f[first].emp_name,f[last].emp_name);
    strcpy(f[last].emp_name,tempchr);
    z++;
}

  void heapify(struct employee * arr, int n, int i) {
    int max = i;
    int leftChild = 2 * i + 1;
    int rightChild = 2 * i + 2;

    if (leftChild < n && arr[leftChild].emp_no > arr[max].emp_no)
      max = leftChild;

    if (rightChild < n && arr[rightChild].emp_no > arr[max].emp_no)
      max = rightChild;

    if (max != i) {
      swap(arr,i,max);
      heapify(arr, n, max);
    }
  }

  void heapSort(struct employee * 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 employee * arr, int n) {
  int i;
  for(i=0;i<n;i++){
    printf("emp_name: %s\n", arr[i].emp_name);
    printf("emp_no: %d\n", arr[i].emp_no);
    printf("emp_salary: %d\n", arr[i].emp_salary);
  }
}

int main() {
  int i, n;
  printf("Enter number of employees: ");
  scanf("%d",&n);
  struct employee arr[n];
  for(i=0;i<n;i++){
      printf("Enter %d emp_no: ", i);
      scanf("%d",&arr[i].emp_no);
      printf("Enter %d emp_name: ", i);
      scanf("%s",&arr[i].emp_name);
      printf("Enter %d emp_salary: ", i);
      scanf("%d",&arr[i].emp_salary);
  }
  printf("Original array:\n");
  display(arr, n);
  heapSort(arr, n);
  printf("Sorted array:\n");
  display(arr, n);
  printf("number of swaps performed are %d\n",z);
}
```

Output given below:

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PS C:\Code\C\Code> cd "c:\Code\C\Code\" ; if ($?) { gcc heapsort.c -o heapsort } ; if ($?) { .\heapsort }
Enter number of employees: 3
Enter 0 emp_no: 9
Enter 0 emp_name: aabb
Enter 0 emp_salary: 4000
Enter 1 emp_no: 5
Enter 1 emp_name: ccdd
Enter 1 emp_salary: 5000
Enter 2 emp_no: 7
Enter 2 emp_name: eeff
Enter 2 emp_salary: 3000
Original array:
emp_name: aabb
emp_no: 9
emp_salary: 4000
emp_name: ccdd
emp_no: 5
emp_salary: 5000
emp_name: eeff
emp_no: 7
emp_salary: 3000
Sorted array:
emp_name: ccdd
emp_no: 5
emp_salary: 5000
emp_name: eeff
emp_no: 7
emp_salary: 3000
emp_name: aabb
emp_no: 9
emp_salary: 4000
number of swaps performed are 3
PS C:\Code\C\Code>