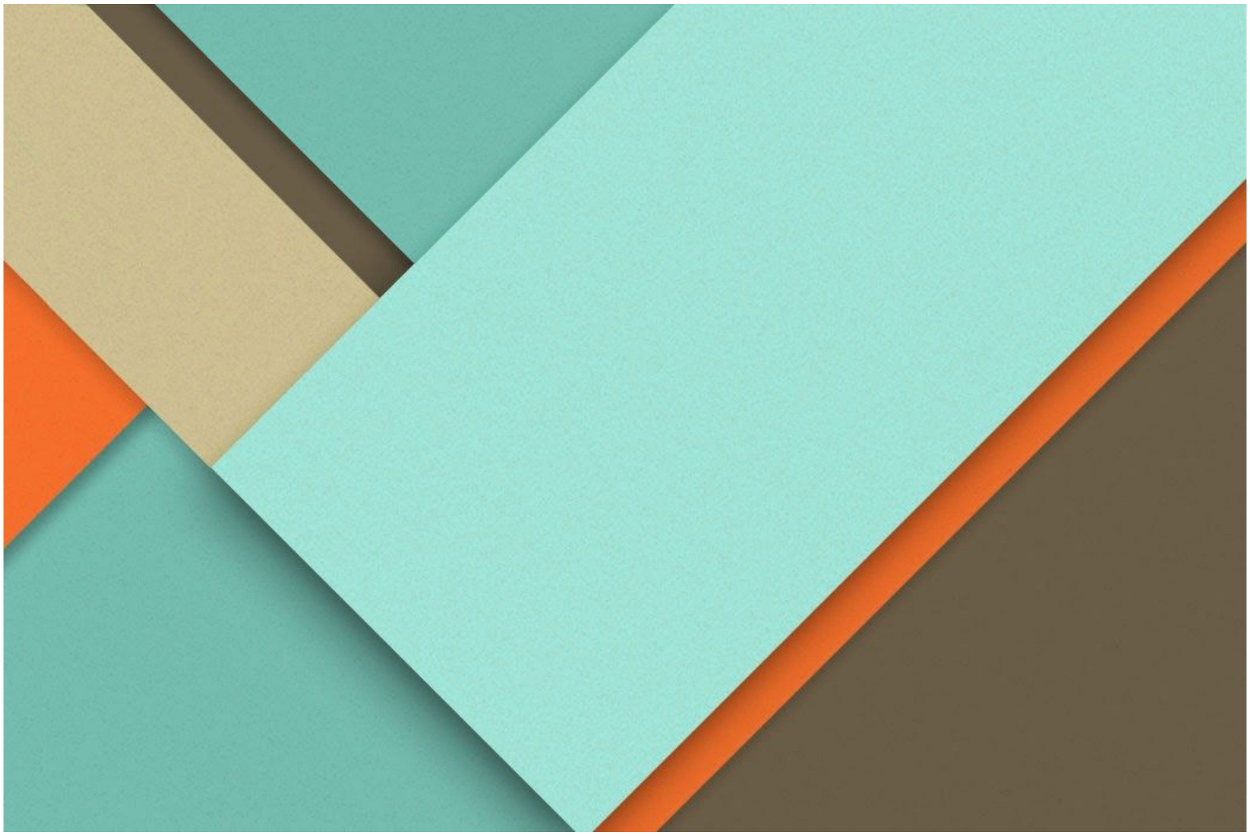

DSA Assignment

Boyapati Sai Venkat

AP19110010174

1st-year CSE-E.

Programs on Searching and Sorting

1. Write a menu-driven program to implement the following operations on an array.

a. Create an integer array of and store N number of integers in the array.

b. Display the content of the array.

c. Linear or sequential search

d. Binary search

Code:

```
#include <stdio.h>
#include<stdlib.h>
int a[100],i,j,n,element,low, high, mid, key, temp;
void create();
void display();
void Lin_search();
void Bin_search();

void main()
{
   int cho;
   do
```

```
    {
        printf("\n.....Menu driven C Program....\n");
            printf("\n 1. Create an integer array\n 2. Display the contents of the arrays\n 3. Linear search\n 4.Binary search\n 5. exit\n");
        printf("Enter your choice");
        scanf("%d", &cho);
        switch (cho)
        {
            case 1: create();
                break;

            case 2: display();
                break;

            case 3: Lin_search();
                break;

            case 4: Bin_search();
                break;

            case 5: exit(0);
                  break;

            default : printf("Enter a valid integer");
        }
    }while(1);
}
```

```
void create()
{
   printf("Enter the value of n ");
   scanf("%d",&n);
   printf("Enter the values in the array\n");
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
}

void display()
{



   printf("The given array is \n");
   for(i = 0 ; i < n ; i++)
   {
      printf("%d\t",a[i]);
   }

}

void Lin_search()
{

   printf("enter the element u want to search\n");
```

```
    scanf("%d", &element);
for(i=0;i<n;++i)
if(a[i]==element)
break;

if(i<n)
printf("Element found at index %d",i);
else
printf("Element not found");
}



void Bin_search()
{
    for(i=1;i<n;i++)
    {
        temp = a[i];
        j = i -1;
    while(j>=0 && a[j]>temp)
    {
        a[j+1] = a[j];
        j = j-1;
    }
    a[j+1] = temp;
    }
    printf("After sorting. Array is :\n");
    for(i = 0 ; i<n ; i++)
```

```
{
    printf("%d\t" , a[i]);
}
printf("\nEnter the element you want to search ");
scanf("%d", &key);
low = 0;
high = n-1;
mid = (low+high)/2;
while(low<= high)
{

    if(a[mid] < key)
    {
        low = mid +1;
    }
    else if(a[mid] == key)
    {
        printf("%d is present at location %d.\n", key , mid+1);
        break;
    }
    else
    {
        high = mid - 1;
    }
    mid = (low + high )/2;
}
if(low>high)
```

```
    {
        printf(" Element %d not found!", key);
    }
}
```

Output:

```
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice1
Enter the value of n 5
Enter the values in the array
24
6
12
143
36

.....Menu-driven C Program...
```

```
 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice2
The given array is
24      6       12      143    36
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice3
enter the element u want to search
6
An element found in index 1
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice3
```

```
enter the element u want to search
15
Element not found
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice4
After sorting. Array is :
6       12      24      36      143
Enter the element you want to search 36
36 is present at location 4.

.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice4
After sorting. Array is :
6       12      24      36      143
Enter the element you want to search 143
```

```
143 is present at location 5.

.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice4
After sorting. Array is :
6       12      24      36      143
Enter the element you want to search 15
 Element 15 not found!
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Linear search
 4.Binary search
 5. exit
Enter your choice5
```

2. Write a menu-driven program to implement the following operations on an array.

a. Create an integer array of and store N number of integers in the array.

b. Display the content of the array.

c. Define a function to implement Insertion sort

d. Define a function to implement Selection sort

e. Define a function to implement Bubble sort

f. Define a function to implement Merge sort

Each of these sorting functions should accept the array and its size as function

parameters.

Code:

```
#include <stdio.h>

#include<conio.h>

#include<stdlib.h>

int a[100],b[100],i,j,k,n,element ;

void create();

void display();

void Bubble_sort (int a[], int n);

void selection_sort(int a[], int n);

void insertion_sort(int a[], int n);

void merge_sort(int a[], int n);



void main()
```

```
{
    int cho;
    do
    {
        printf("\n.....Menu driven C Program....\n");
            printf("\n 1. Create an integer array\n 2. Display the contents of the arrays\n 3. Insertion sort\n 4.Selection sort\n 5.Bubble sort\n 6.Merge sort\n 7. exit\n");
        printf("Enter your choice");
        scanf("%d", &cho);
        switch (cho)
        {
            case 1: create();
                break;

            case 2: display();
                break;

            case 3: insertion_sort(a, n);
                break;

            case 4: selection_sort(a, n);
                break;

            case 5 : Bubble_sort( a, n);
                break;

            case 6: printf("After merge sort. Array is : \n");
                    for(i=0 ; i<n ; i++)
                    {
                        printf("%d\t" , a[i]);
                    }
                    merge_sort(a, n);
                    break;

            case 7: exit(0);
                  break;

            default : printf("Enter a valid integer");
        }
    }while(1);
}

void create()
{
    printf("Enter the value of n ");
    scanf("%d",&n);
    printf("Enter the values in the array\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
}

void display()
{
    printf("The given array is  \n");
    for(i = 0 ; i < n ; i++)
    {
        printf("%d\t",a[i]);
    }

}

void Bubble_sort(int a[], int n)
{
    int temp;
    for(i = 0 ; i<n-1 ; i++)
    {
        for(j = 0 ; j < n-1 ; j++)
        {
            if(a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp ;
            }
        }
    }
    printf("After Bubble sort. Array is : \n ");
    for(i=0;i<n;i++)
    {
        printf("%d \t ", a[i]);
    }
}


void selection_sort(int a[], int n)
{
    int min, temp;

    for(i = 0 ; i<n-1; i++)
    {
        min = i;
        for(j = i+1 ; j<n; j++)
        {
            if(a[j]<a[min])
            {
                min = j;
            }
        }
        if(min != i)
        {
            temp = a[i];
            a[i] = a[min];
            a[min] = temp;
        }
    }
    printf("After selection sort. Array is : \n ");
    for(i= 0 ; i<n ; i++)
    {
        printf("%d\t ", a[i]);
    }
}



void insertion_sort(int a[], int n)
{
    int temp;
    for(i=1;i<n;i++)
    {
        temp = a[i];
        j = i -1;
    while(j>=0 && a[j]>temp)
    {
        a[j+1] = a[j];
        j = j-1;
    }
    a[j+1] = temp;
    }
    printf("After insertion sort. Array is :\n");
    for(i = 0 ; i<n ; i++)
    {
        printf("%d\t" , a[i]);
    }
```

```
}

void merge_sort(int a[] , int n)
{
    int Low_bound, up_bound, mid;
    i = Low_bound;
    j = mid + 1;
    k = Low_bound;
    while(i<=mid && j<= up_bound)
    {
        if(a[i]<=a[j])
        {
            b[k] = a[i];
            i++;
        }
        else
        {
            b[k] = a[j];
            j++;
        }
        k++;
    }
    if(i>mid)
    {
        while(j<=up_bound)
        {
            b[k]=a[j];
            j++;
            k++;

        }
    }
    else
    {
        while(j<= mid)
        {
            b[k] = a[i];
            i++;
            k++;
        }
    }
    for(k = Low_bound ; k<=up_bound ; k++)
    {
        a[k] = b[k];
    }
}
```

Output:

```
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice1
Enter the value of n 2 6
Enter the values in the array
1
4
2
4
6
3

.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice2
The given array is
1       4       2       4       6       3
.....Menu-driven C Program...
```

```
 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice3
After insertion sort. Array is :
1       2       3       4       4       6
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice2
The given array is
1       2       3       4       4       6
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice4
After selection sort. Array is :
 1       2       3       4       4       6
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice2
The given array is
1       2       3       4       4       6
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice5
After Bubble sort. Array is :
 1       2       3       4       4       6
.....Menu driven C Program....

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice2
The given array is
1       2       3       4       4       6
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice6
```

```
After the merge sort. Array is :
1       4       2       4       6       3
.....Menu-driven C Program...
 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice2
The given array is
1       4       2       4       6       3
.....Menu-driven C Program...

 1. Create an integer array
 2. Display the contents of the arrays
 3. Insertion sort
 4.Selection sort
 5.Bubble sort
 6.Merge sort
 7. exit
Enter your choice7
Exit.
```