PORT CITY INTERNATIONAL UNIVERSITY

Lab report on program code using specific Algorithms .

Course Title : Algorithms Sessional

Course Code : CSE 228

Submitted To :

Md. Hasanuzzaman Noor

Lecturer, Dept. of CSE

Port City International University

Chittagong

Submitted By :

Name – Anik Dhar

ID – CSE 00405170

Program – B.Sc. in CSE

Batch – 004[Day]

Remarks :

Lab Report no. : 01

Write a program for searching an element with the Binary Search algorithm .

Objectives :

The main purpose of this algorithm that how to search an element in a array , using the Binary Search algorithm .

Equipments :

- A Desktop / Laptop Computer
- Operating System : Windows 7 , 8 , 10 / Ubuntu Linux : 14.10 / 15.10
- Platform : C
- IDE : Code::Blocks / Notepad editor
- Compiler : GNU GCC for C

Description :

To complete the given task of writing a program about searching an element from a certain array following steps are taken :

- The first condition is a sorted array . If we have to use the binary search algorithm , then the certain array has to be sorted .
- The searching will take place by going first to the middle position . To get the middle position we have to divide the whole array in to 3 sectors . Beg , End and Mid . And Mid = (Beg+End)/2 .
- From the mid position it will check if the searched element is greater than or less than or equal to mid .
- If the element is greater , then the searching will go to the right side of the array . If it is less , then the searching will go to the left side of the array . This process will continue to go on until the desired element .
- If the element is equal to the mid , then the desired element is found and it will show the result .
- The time complexity of binary search algorithm is O(log n).

Source Code :

```
#include<stdio.h>

int main()

{

   int i,n,first,middle,last,search,array[100];
```

```
printf("Enter desired number of elements : ");
scanf("%d",&n);

printf("\nEnter %d integers : ",n);

    for(i=0;i<n;i++)
    {
    scanf("%d",&array[i]);
    }
    printf("\nEnter value to search : ");
    scanf("%d",&search);

    first=0;
    last=n-1;
    middle=(first+last)/2;

    while(first<=last && search!=array[middle])
       {
       if(search>array[middle])
          first=middle+1;

       else
          last = middle-1;
          middle=(first+last)/2;
       }
    if(search==array[middle])
    {
```

```
            printf("\n%d Found at location %d \n",search,middle+1);


        }
        if(first>last)
        {
            printf("Searched number not found! %d is not present in the list.\n",search);
        }
        getch();
}
```

Output :

```
"C:\Users\Anik Dhar Papon\Documents\Binary_Search.exe"
Enter desired number of elements : 10

Enter 10 integers :

1       2       3       4       5       6       7       8       9       10

Enter value to search : 7

7 Found at location 7

Process returned 13 (0xD)   execution time : 22.705 s
Press any key to continue.
```

Discussion :

- I had to write the code carefully , in order to prevent syntax error .
- From this experiment I have learnt how to write C code in Linux Ubuntu .
- To execute the program correctly I had to make sure the command lines are correct .

Lab Report no. : 02

Write a program for sorting the array element with the Insertion Sort algorithm .

Objectives :

The main purpose of this algorithm that how to sort the elements in an unsorted array , using the Insertion Sort algorithm .

Equipments :

- A Desktop / Laptop Computer
- Operating System : Windows 7 , 8 , 10 / Ubuntu Linux : 14.10 / 15.10
- Platform : C
- IDE : Code::Blocks / Notepad editor
- Compiler : GNU GCC for C

Description :

To complete the given task of writing a program about sorting the element from a certain array following steps are taken :

- The first condition is a the array has to be unsorted and the sorting portion will start from the second element of the array .
- Here the code of the insertion sort is consists of several blocks . Initialization , Computation , Termination .
- In Initialization we take the input value of the array .
- In the computation section we sort the whole array using the Insertion Sort algorithm .
- Finally in termination we terminate the program by showing the result of the sorted array.
- The time complexity of Insertion Sort algorithm is O(n^2).

Source Code :

```
#include<stdio.h>

int main()

{

   int n,i,j,temp,num[100];


   printf("\nEnter the desired number you want to print:");

   scanf("%d",&n);
```

```
printf("\nInclude %d numbers in the array:\n",n);

printf("\nThe numbers are:");
for(i=0;i<n;i++)
{
   scanf("%d",&num[i]);
}

printf("\nInitially the values are: ");
for(i=0;i<n;i++)
{
   printf("\nElement at %d position is %d",i,num[i]);
}

for(i=1;i<=n-1;i++)
{
   while(i>0 && num[i]<num[i-1])
   {
      temp=num[i];
      num[i]=num[i-1];
      num[i-1]=temp;
      i--;
   }
}


printf("\nThe Sorted numbers in ascending order are: ");
```

```
    for(i=0;i<n;i++)
    {
        printf(" %d",num[i]);


    }
}
```

Output :

```
"C:\Users\Anik Dhar Papon\Documents\Insertion_Sort.exe"

Enter the desired number you want to print: 10

Include 10 numbers in the array:

The numbers are:
8       1       9       2       0       4       6       7       8       3

Initially the values are:
Element at 0 position is 8
Element at 1 position is 1
Element at 2 position is 9
Element at 3 position is 2
Element at 4 position is 0
Element at 5 position is 4
Element at 6 position is 6
Element at 7 position is 7
Element at 8 position is 8
Element at 9 position is 3
The Sorted numbers in ascending order are:  0 1 2 3 4 6 7 8 8 9
Process returned 10 (0xA)   execution time : 35.733 s
Press any key to continue.
```

Discussion :

- I had to write the code carefully , in order to prevent syntax error .
- From this experiment I have learnt how to write C code in Linux Ubuntu .
- To execute the program correctly I had to make sure the command lines are correct .

Lab Report no. : 03

Write a program for sorting the array element with the Merge Sort algorithm.

Objectives :

The main purpose of this algorithm that how to sort the elements in an unsorted array , using the Merge Sort algorithm .

Equipments :

- A Desktop / Laptop Computer
- Operating System : Windows 7 , 8 , 10 / Ubuntu Linux : 14.10 / 15.10
- Platform : C
- IDE : Code::Blocks / Notepad editor
- Compiler : GNU GCC for C

Description :

To complete the given task of writing a program about sorting the element from a certain array following steps are taken :

- Here we have done this program by using recursion method .
- Here the code of the merge sort is consists of several blocks . The most important are Divide & Conquer .
- After the divide and conquer portion of the program we used Combine portion to run the code .
- In this sorting algorithm , we divide the problem in small pieces , then solve those small pieces and combine the solutions we get the final result .

Source Code :

```
#include<stdio.h>

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

void merge_sort(int a[], int p, int r)

{

   int q;

   if(p<r)

   {

      q=(p+r)/2;

      merge_sort(a,p,q);

      merge_sort(a,q+1,r);
```

```
        merge(a,p,q,r);
    }
}
void merge(int a[], int p, int q, int r)
{
    int n1,n2,lt[50],rt[50],i,j,k;
    n1=q-p+1;
    n2=r-q;
    for(i=1;i<=n1;i++)
    {
        lt[i]=a[p+i-1];
    }
    for(j=1;j<=n2;j++)
    {
        rt[j]=a[q+j];
    }
    lt[n1+1]=10000;
    rt[n2+1]=10000;
    i=1;
    j=1;

    for(k=p;k<=r;k++){
        if(lt[i]<=rt[j])
        {
            a[k]=lt[i];
            i++;
        }
        else
```

```
        {
            a[k]=rt[j];
            j++;
        }

    }
}
int main()
{
    int p,r,i,x,n;
    int a[50],lt[50],rt[50];
    printf("\nEnter the array size: ");
    scanf("%d",&n);
    printf("\nEnter %d element: ",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    p=0;
    r=n-1;

   merge_sort(a, p, r);
   printf("\nThe sorted array is: \n");
   for(x=0;x<n;x++)
   {
      printf("%d\t",a[x]);
   }
    return 0;
```

}

Output :


```
Enter the array size: 10

Enter 10 element:
1       9       2       10      4       8       6       5       3       7

The sorted array is:
1       2       3       4       5       6       7       8       9       10
Process returned 0 (0x0)   execution time : 25.032 s
Press any key to continue.
```

Discussion :

- I had to write the code carefully , in order to prevent syntax error .
- From this experiment I have learnt how to write C code in Linux Ubuntu .
- To execute the program correctly I had to make sure the command lines are correct .