314455: SOFTWARE LABORATORY – V

Teaching Scheme: **Examination Scheme:**
Practicals : 4 Hours/Week **Termwork : 50 Marks** **Practical : 50 Marks**

GROUP A

1. Write a program to implement Pass-I of Two-pass assembler for Symbols and Literal processing considering following cases:
    i. Forward references
    ii. DS and DC statement
    iii. START, EQU, LTORG, END.
    iv. Error handling: symbol used but not defined, invalid instruction/register etc.
2. Write a program to implement Pass-II of Two-pass assembler for output of Assignment 1.
3. Study Assignment for Macro Processor. (Consider all aspects of Macro Processor)
4. Write a program to implement Lexical Analyzer for subset of C.
5. Write a program to implement a Recursive Descent Parser.
6. Write a program to implement calculator using LEX and YACC.
7. Write a program for Intermediate code generation using LEX &YACC for Control Flow statement (Either While loop or Switch case)

GROUP B

1. Write a program to find Maximum and Minimum element in an array using Divide and Conquer strategy and verify the time complexity.
2. Write a program to solve optimal storage on tapes problem using Greedy approach.
3. Write a program to implement Bellman-Ford Algorithm using Dynamic Programming and verify the time complexity.
4. Write a program to solve the travelling salesman problem and to print the path and the cost using Dynamic Programming.
5. Write a recursive program to find the solution of placing n queens on chessboard so that no two queens attack each other using Backtracking.
6. Write a program to solve the travelling salesman problem and to print the path and the cost using Branch and Bound.

Name: Aditya Somani | TE (IT) – A | Roll No. T1851061 | PRN No. 71901204L

Assignment No. 1

AIM: Write a program to find Maximum and Minimum element in an array using Divide and Conquer strategy and verify the time complexity.

OBJECTIVE:

1. To understand the concept of Divide and Conquer.
2. To find Maximum and Minimum element in an array.
3. To find Time Complexity of algorithms

THEORY:

What is Divide and Conquer?

Divide-and-conquer is a top-down technique for designing algorithms that consists of dividing the problem into smaller subproblems hoping that the solutions of the subproblems are easier to find and then composing the partial solutions into the solution of the original problem.

Little more formally, divide-and-conquer paradigm consists of following major phases:

- Divide: Breaking the problem into several sub-problems that are similar to the original problem but smaller in size,
- Conquer: Solve the sub-problem recursively (successively and independently), and then
- Combine: Combine these solutions to subproblems to create a solution to the original problem.

Advantages

1. Solving difficult problems
2. Algorithm efficiency parallelism
3. Memory access
4. Roundoff control

Maximum and Minimum element in an array

In this approach, the array is divided into two halves. Then using recursive approach maximum and minimum numbers in each halves are found. Later, return the maximum of two maxima of each half and the minimum of two minima of each half.

In this given problem, the number of elements in an array is y−x+1, where y is greater than or equal to x.

The function Maxmin(x,y) will return the maximum and minimum values of an array numbers[x...y].

Complexity Analysis

Let T(n) be the number of comparisons made by Maxmin(x,y), where the number of elements n=y−x+1.

If T(n) represents this no., then the resulting recurrence relations is

$$T(n)=T([n/2]+T[n/2]+2 \quad n>2$$

$$1 \quad n=2$$

$$0 \quad n=1$$

So time complexity is O(n) after solving above recurrence equation.

PROGRAM

```
#include<stdio.h>

int max, min;
int a[100];

void maxmin(int i, int j)
{
   int max1, min1, mid;

   if(i==j)
            max = min = a[i];
   else
   {
            if(i == j-1)
            {
                     if(a[i] <a[j])
                     {
                              max = a[j];
                              min = a[i];
                     }
                     else
                     {
                              max = a[i];
                              min = a[j];
```

```
                }
            }
            else
            {
                mid = (i+j)/2; maxmin(i, mid);
                max1 = max; min1 = min; maxmin(mid+1, j); if(max <max1)
                max = max1; if(min > min1) min = min1;
            }
    }
}

int main ()
{
    int i, num;
    printf ("\nEnter the total number of numbers : ");
    scanf ("%d",&num);
    printf ("Enter the numbers : \n");
    for (i=1;i<=num;i++)
    scanf ("%d",&a[i]);

    max = a[0];
    min = a[0];
    maxmin(1, num);
    printf ("Minimum element in an array : %d\n", min);
    printf ("Maximum element in an array : %d\n", max);
    return 0;
}
```

output:-

```
Enter the total number of numbers : 6
Enter the numbers :
3
45
7
1
12
64
Minimum element in an array : 1
Maximum element in an array : 64

--------------------------------
Process exited after 23.67 seconds with return value 0
Press any key to continue . . .
```

Conclusion:

Here we find the maximum and minimum elements of an array by using Divide and conquer approach. The complexity is also calculated.