Apex College

BCIS Program

Affiliated to Pokhara University

APEX COLLEGE
Affiliated to Pokhara University
Building Human Capital.

Data Structure & Algorithms
Lab Report

4. Implementation of Recursion on Factorial, Fibonacci & Tower of Hanoi

Date: 17-05-2022

Submitted by:
Ishwor Shrestha
Roll no.: 2018-BCIS-414

Submitted to:
Pravakar Ghimire, &
Anmol Shrestha
Apex College

CS CamScanner

#Lab4 Objectives

- To understand recursion
- To implement recursive functions
- To findout factorial, fibonacci series and tower of hanoi problem using recursive function.

#Introduction

- Recursion is a way of defining problem on its own term. It is a method of solving a computational problem where the solution depends on solutions to smaller instances of same problem.
- It is a Buttom up approach to solve problem.
- Recursive function implicitly uses the stack.
- In recursive function, there must be base condition to find recursive output.
- We can use iterative method to solve all the problems that can be solve using recursive function.
- In recursion, we call function inside function which needs to switch context, so that it results by taking more time.
- It has O(n) time complexity, but we still use recursion because of it's simplicity to write algorithm ~~an~~ and solve large problems in few steps.

*problems that we can solve using recursion:

1. Factorial
2. Fibonnacci Series
3. Adding a list of numbers
4. Sudoku
5. Tower of Hanoi

#programs to implement Recursion in Factorial, Fibonacci sequence and Tower of Hanoi.

* For Factorial

$n! = n * (n-1) * (n-2) * \dots * 1! * 0!$

```
#include <stdio.h>

~~factorial~~
int factorial (int n) {
    if (n==0)
        return 1;

    else
        return n * factorial (n-1);
}
void main() {
    int n;
    printf("Enter value of n':");
    scanf("%d", &n);
    printf("Factorial of %d is %d. \n", n, factorial(n));
}
```

* For Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 44, ... n

```
#include <stdio.h>

int fibonacci (int n) {
    if (n==0 || n == 1)
        return n;

    else
        return (fibonacci(n-1) + fibonacci (n-2));
}

int main() {
    int n, n=0;
    printf("Enter a number of terms in Fibonacci Sequence.\n");
    scanf("%d", &n);
```

CS CamScanner

```
for (int i=1; i<=n; i++){
    printf("%d \t", fibonacci (n));
    n++;
}

return 0;
}
```

*For Tower Of Hanoi

```
#include<stdio.h>
void toh (int n, char s, char t , char d) {
    if (n>0) {
        toh (n-1 , s, d, t);
        printf("move disk %d form %c to %c.\n", n, s, d);
        toh (n-1, t, s, d);
    }
}

void main (){
    int n;
    printf("Enter the number of disk on source Tower: ");
    scanf("%d", &n);

    toh(n, 'S', 'T', 'D');
}
```

#Activities

To understand the concept of recursion, we have practice three different problems;

① Factorial of nth number

1. Get user input at variable 'n'
2. Check, if n is 0, then print ~~&~~ return 1.

```
    Otherwise, return n * factorial (n-1)
3. End
```

② Fibonacci Series of nth term

```
1. Get number of terms in Fibonacci sequence.
2. Start loop to contineous iteration from 1 to n by 1 increment
       i.e. for (int i=0 ; k<=n ; i++){
                printf("%d \t", fibonacci(m));
                m++;
            }
3. Check if n terms is 0 or 1, then return value of n
4. Otherwise, return (fibonacci (n-1) + fibonacci(n-2));
5. End
```

③ Tower of Hanoi

```
1. Check if n=1;
     - move a single disk from source to destination
2. If, n>1:
     - Let, temp be remaining pole other than source
       and destination
     - move a tower of (n-1) disks from source to temp tower
     - move a single disk from source to destination
     - move a tower of (n-1) disk from temp to destination
3. End
```

Conclusion

In lab 4, I have learned about the implementation of recursion in terms of factorial problem, fibonacci series problem and tower of hanoi solution.

CS CamScanner