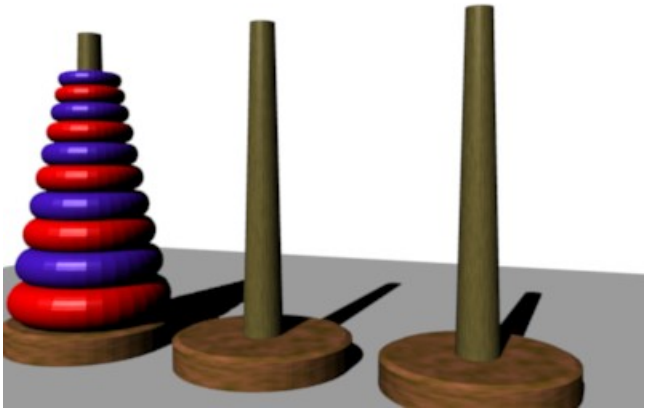
Universitatea din Craiova
Facultatea de Automatică,Calculatoare şi Electronică

01 June 2018



Project : Programming Techniques
Title : A library for matrix functions
Teachers : Becheru Alex & Bădică Costin & Murareţu Ionuţ
Student : Negrea Andrei Adelin
Section : Calculatoare Română
Year I
Group : 1.2 B

Contents

1 Problem Statement **3**
1.1 Title . . . 3
1.2 Description . . . 3

2 Pseudocode **4**
2.1 Read File . . . 4
2.2 Divide et Impera . . . 4
2.3 Stack Implementation . . . 4

3 Application Design **6**
3.1 Recursiv function . . . 6
3.2 Input Data . . . 6
3.3 Output Data . . . 6
3.4 Functions . . . 6

4 Source Code **6**

5 Experiments and results **7**
5.1 GNU GCC Compiler . . . 7

6 Conclusion **8**

7 References **9**

1 Problem Statement

1.1 Title

Tower of Hanoi

1.2 Description

My project refers to an application that will solve the hanoi towers problem. There are three given rods defined as A,B,C. On rod A we have n plates of different diameters,in increasing order of diameters,from top to bottom. Initially rods B and C are empty. The application will display all the moves in the next order: plates on the rod A will be moved to rod B,in the same order,using C as a maneuvering rod and respect the following rules: at every step one plate will move; a disk can only be placed over a larger diameter disc.

2 Pseudocode

2.1 Read File

```
1: START
2: INT i, j, n, m
3: FILE * f
4: f = fopen("date.out", "a")
5: fseek(f, nr * 2, SEEK_CUR)
6: fprintf(f, "%d- > %d", a, b)
7: print("%d- > %d", a, b)
8: fclose(f)
```

2.2 Divide et Impera

```
if n then
    hanoi(n − 1, a, c, b)
    write(a, b)
    nr + +
    hanoi(n − 1, c, b, a)
end if
```

2.3 Stack Implementation

```
START
INT i
FILE * out
out = fopen("output.txt", "a")
for i = 1 to n do
    if i > 0 then
        if i%2 then
            if t%2 then
                s2 = y
                if stack[i]! = 0 then
                    r2 = stack[i] − 1
                else
                    r2 = s2
                end if
                if r2 < 1 then
                    r2 = 3
                end if
            end if
        else
            s2 = y − 1
            if stack[i]! = 0 then
                r2 = stack[i] + 1
```

```
        else
          r2 = s2
        end if
        if r2 > 3 then
          r2 = 1
        end if
        stack[i] = r2
        printf("Disk%dmovefrom%dto%d", i, x, r2)
        fprintf(out, "Disk%dmovefrom%dto%d", i, x, r2)
        f(i − 1, r1, r2)
      end if
    else
      if t%2 then
        s1 = y − 1
      end if
      if stack[i]! = 0 then
        r1 = stack[i] − 1
      else
        r1 = s1
      end if
      if r1 < 1 then
        r1 = 3
      end if
      stack[i] = r1
      printf("Disk%dmovefrom%dto%d", i, x, r1)
      fprintf(out, "Disk%dmovefrom%dto%d", i, x, r1)
      f(i − 1, r2, r1)
    end if
end for
End
```

3 Application Design

3.1 Recursiv function

In my problem :

If n = 1, A− >B move is made,namely the plate on the rod A moves on rod B. If n = 2, A− >C , A− >B, C− >B moves are made. In case of n > 2 , the problem is complicated. We will note H(n,a,b,c) the sequence of the n drives moves from rod A to rod B using as a intermediate rod, rod C. According to the Divide et Impera strategy, we will try to break the problem in two other subproblems of the same type, then we will combine the solutions. In this case, we observe that the move of the n plates is equivalent to:
- n-1 move from rod A to rod C, using rod B
- moving the remaining disk on rod B
- n-1 move from rod C to rod B, using rod A

3.2 Input Data

For my problem the input will be n. N is the number of disks and it's introduce from the keyboard in the runscreen.

3.3 Output Data

The output data which represents the actual result of the problem step by step will be present in the file output_method1.txt for the first algorithm and respectively output_method2.txt for the second algorithm .

3.4 Functions

The functions used in the program are presented in the **Section 2**, in their pseudocode forms.

4 Source Code

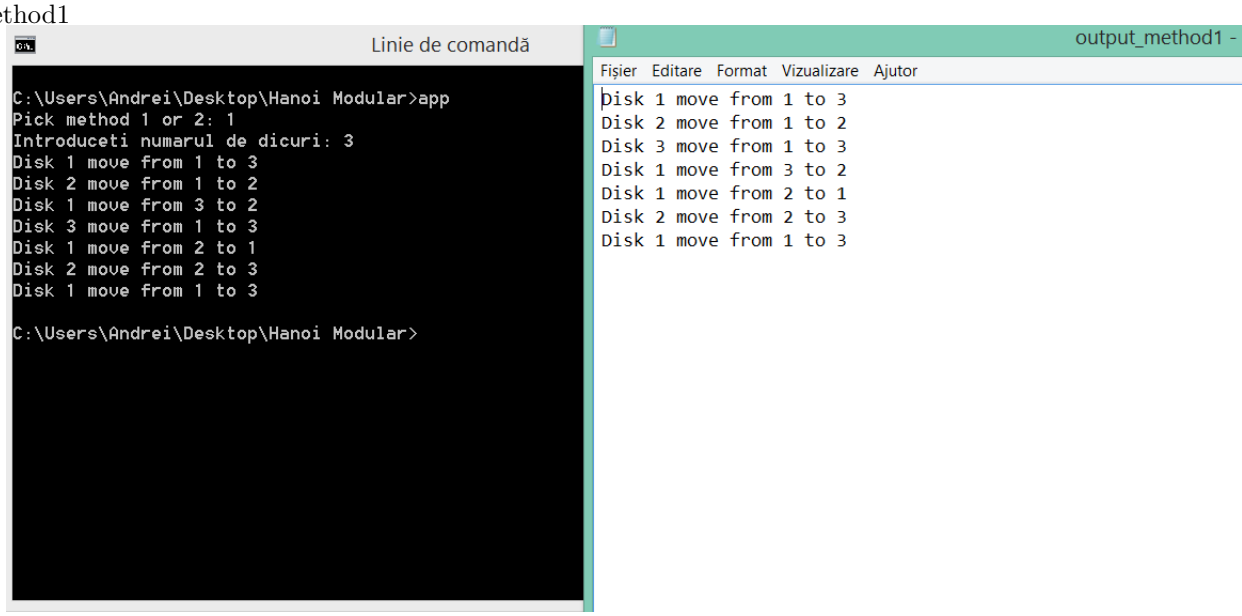
My project is called "Tower of Hanoi." . The source code is created in programming language standard C99, that is compiled in one compiler.

The compiler is in GNU GCC Compiler with help program Code Blocks 16.

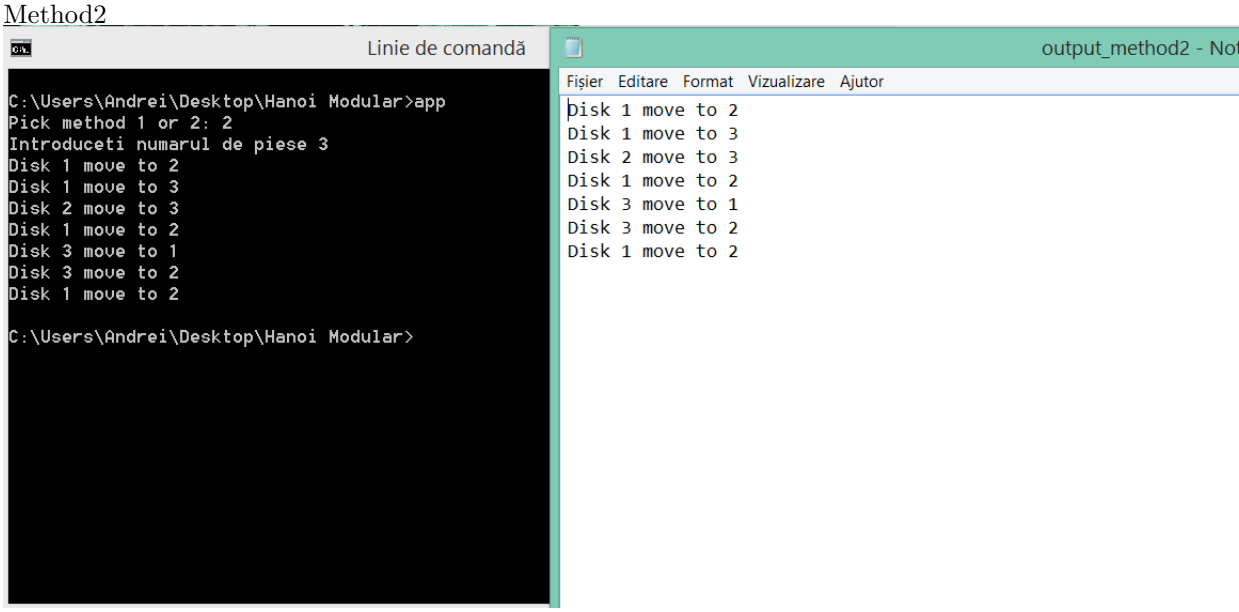
5 Experiments and results

5.1 GNU GCC Compiler

Method1



Method2



6 Conclusion

This program depends very much of the number of the plates introduced by the user. If we will have a number of towers larger than 15 the execution time will be bigger

7 References

Book:
Name : Totul despre C si C++
Year of publication :2005
Publisher :Teora
Author :Dr. Kris Jamsa Lars Klander
Web references:
1.$http://www.geeksforgeeks.org$
2.$https://www.sharelatex.com/learn/Main_Page$
Article:
1.$https://divideetimpera.wikispaces.com/Turnurile+din+Hanoi$