Project for KIV/UPA

Task 02 - Program in MIPS

Jakub Šilhavý
A17B032P
silhavyj@students.zcu.cz

22. 10. 2020

Contents

1 Assignment description

The program will print out a step by step solution for the problem of Hanoi Towers where the number of disks on the first rod is between 1 and 9.

2 Analysis

2.1 Problem description

A detailed description of this problem can be found at `https://en.wikipedia.org/wiki/Tower_of_Hanoi`.

Tower of Hanoi is a mathematical problem where disks of different diameters are being moved among three rods. In the initial configuration, all the disks are stacked on the first rod such that the largest disk is placed at the bottom and the smallest one on the top. The goal is to efficiently move all the disks from the first rod to the last one using the middle rod. In other words, all the disks from the first rode must be put onto the last one in the same order with as minimal moves as possible. Additionally, there are some rules that must be followed (2.2).

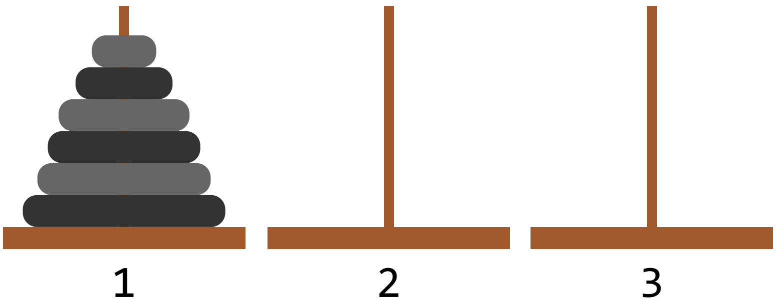

Figure 2.1: Visualization of the problem

2.2 Rules

(1) Only one disk can me moved at a time.

(2) One move means taking a disk off the top of a rod a placing it on the top of another.

(3) It is not allowed to place a disk of a larger diameter onto a disk of a smaller diameter.

3 Algorithm

The input of the procedure solving this problem would be `N, R1, R2, and R3`, where `N` is the number of disks and `R1, R2, and R3` are the numbers of rods. We want to move `N` disks from rod `R1` to rod `R2`.

The solution is trivial for `N = 1`. We simply move the disk from rod `R1` to rod `R2` and we are all done. In case `N > 1`, we need to break the whole algorithm down into smaller steps. It is clear that we need to first move `N-1` disks from rod `R1` to rod `R3`, as a temporary step, so we can move the largest disk from rod `R1` right onto rod `R2`. Once the largest disk is at its final position, all we need to do is put all the `N-1` disks from rod `R3` onto rod `R2` and the problem is solved.

Now we are facing another problem - how to move `N-1` disks from rod `R1` to rod `R3`. We can simply use the same idea and think of rod `R3` as rod `R2` instead. We can break this down further and further until we are left with moving only one disk.

3.1 Pseudo code

Algorithm 1 Hanoi Towers

procedure HANOI(N, R1, R2, R3)
    if $N == 1$ **then**
        print *R1 -> R2*
        return
    hanoi$(N-1, R1, R3, R2)$ ▷ move $N-1$ disks from R1 to R3
    hanoi$(1, R1, R2, R3)$ ▷ move the largest disk from R1 to R2
    hanoi$(N-1, R3, R2, R1)$ ▷ move $N-1$ disks from R3 to R2

3.2 Time complexity

The algorithm is recursive and works with a time complexity of $O(2^n)$, where n is the number of disks on the first rod. An example of how many moves it takes for a specific number of disks is shown in the table below.

Table 3.1: Example of particular number of disks

Number of disks	**Total number of moves**
1	1
2	3
3	7
4	15
10	1023
20	1048575
25	33554431
60	$1,1529.10^{18}$
100	$1,2677.10^{30}$
N	2^{N-1}

4 Implementation

4.1 Description

First, the program encourages the user to enter a number of disks, for which they want to print all the moves solving the problem of Hanoi Towers. If the user enters an invalid number that is not between 1 and 9, the program will print out an error message and terminate. Otherwise, it will print out all the moves for the particular number of disks the user entered.

4.2 Example of I/O

4.2.1 Example A

```
Enter the number of disks:  4
1 -> 3
1 -> 2
3 -> 2
1 -> 3
2 -> 1
2 -> 3
1 -> 3
1 -> 2
3 -> 2
3 -> 1
2 -> 1
3 -> 2
1 -> 3
1 -> 2
3 -> 2
```

4.2.2 Example B

```
Enter the number of disks:  -5
The number is supposed to be between 1 and 9!
```

5 Conclusion

The algorithm was implemented using the Assembly language MISP and tested in Mars simulator - `http://courses.missouristate.edu/kenvollmar/mars/`. Everything works as required, and furthermore, an input validation was implemented within the program as well.