University of Deusto

Inteligent System

Hanoi puzzle

Authors:
Diana Artiom
Victor Luca

Professor:
Enrique Onieva

April 11, 2017

1 Introduction

1.1 Initial configuration

The Tower of Hanoi is a famous puzzle which is might be described as follows:

- It consists of three rods, and a number of disks of different sizes which can slide onto any rod.
- The puzzle starts with the disks in a neat stack in ascending order of size on the leftmost rod, the smallest at the top, thus making a conical shape.

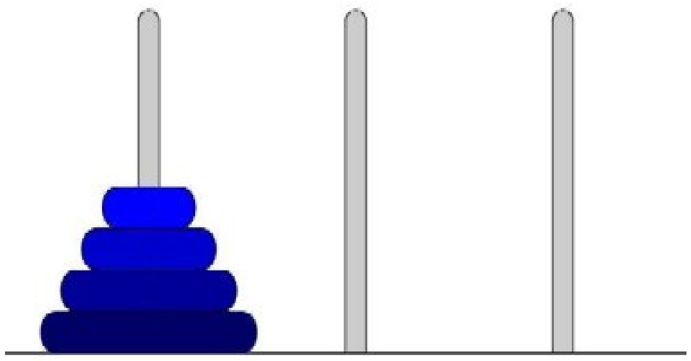


Figure 1: Hanoi tower initial configuration

1.2 Purpose of the puzzle

The objective of the puzzle is to move the entire stack to the rightmost rod obeying the following rules:

- Only one disk may be moved at a time
- Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks
- No disk may be placed on top of a smaller disk that may already be present on that rod.

2 Solution

The minimal number of moves required to solve a Tower of Hanoi puzzle $is2^n - 1$, where n is the number of disks. In order to solve hanoi puzzle we have to codify it first. We can easily do that using a simple vector of length k corresponding to number of roads in puzzle. For example configuration from the Figure 1 might be might be codified as follows:

$$initialState < 1, 1, 1, 1 >$$

The configuration above is the one used for initial state. For the final state configuration will look like this:

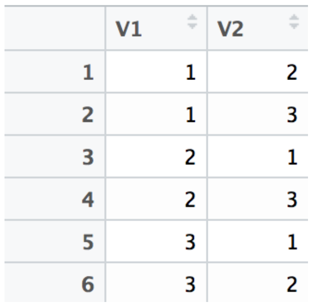
$$finalState < 3, 3, 3, 3 >$$

Any move of the disk from one rod to another is codified with a vector with two values:

$$move(1, 2)$$

So, the move presented above is read: "Move from rod 1 to rod 2".

There is one more thing to be added in order to have complete setup for this puzzle and that is possible actions of the move. Possible actions are all combinations of moves to be or not to be applied.Having a configuration of 4 disks and 3 rods possible actions are:

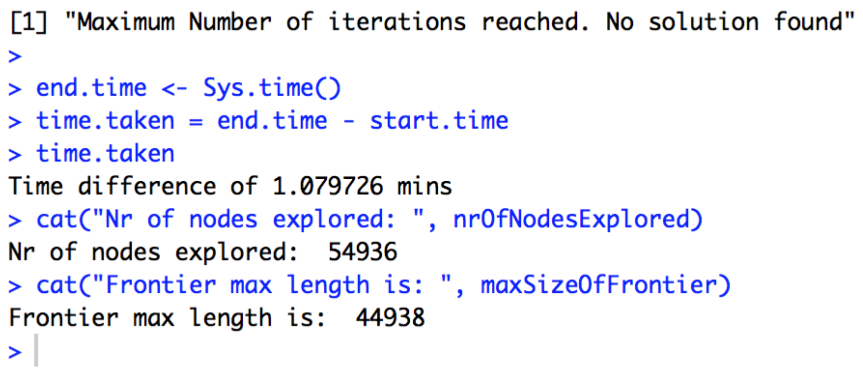


	V1	V2
1	1	2
2	1	3
3	2	1
4	2	3
5	3	1
6	3	2

Figure 2: Possible actions

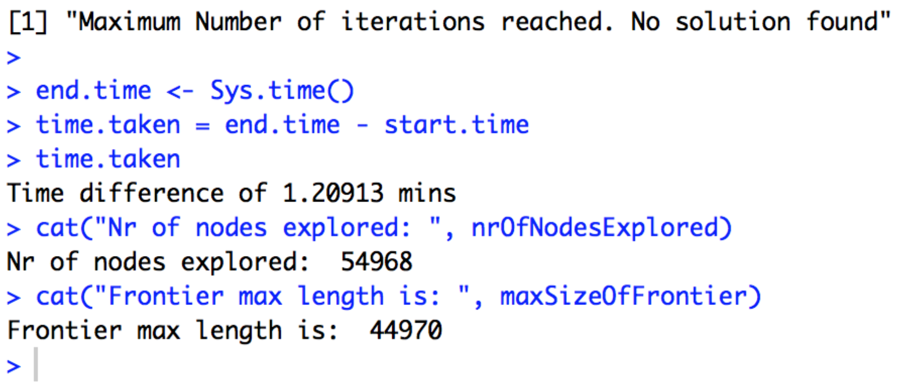
Now, since we have our codification of the problem set up and ready, we can solve hanoi by expending the tree of possible action.

3 Result analysis



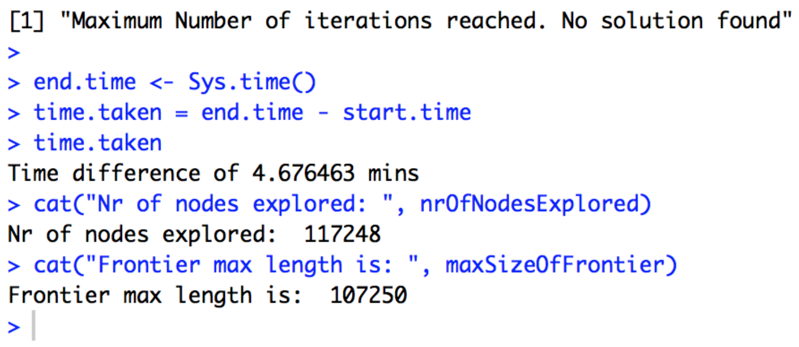
```
[1] "Maximum Number of iterations reached. No solution found"
>
> end.time <- Sys.time()
> time.taken = end.time - start.time
> time.taken
Time difference of 1.079726 mins
> cat("Nr of nodes explored: ", nrOfNodesExplored)
Nr of nodes explored:  54936
> cat("Frontier max length is: ", maxSizeOfFrontier)
Frontier max length is:  44938
>
```

Figure 3: Configuration - 4 rods and 4 disks



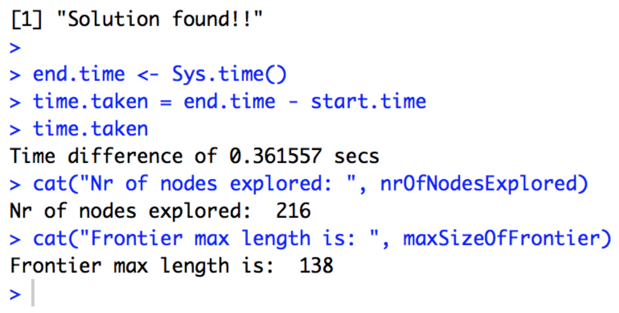
```
[1] "Maximum Number of iterations reached. No solution found"
>
> end.time <- Sys.time()
> time.taken = end.time - start.time
> time.taken
Time difference of 1.20913 mins
> cat("Nr of nodes explored: ", nrOfNodesExplored)
Nr of nodes explored:  54968
> cat("Frontier max length is: ", maxSizeOfFrontier)
Frontier max length is:  44970
>
```

Figure 4: Configuration - 4 rods and 6 disks



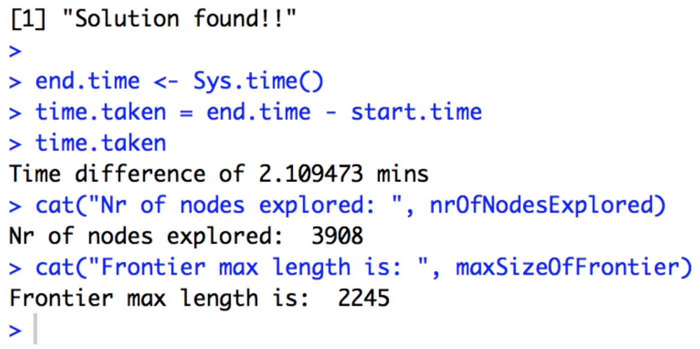
```
[1] "Maximum Number of iterations reached. No solution found"
>
> end.time <- Sys.time()
> time.taken = end.time - start.time
> time.taken
Time difference of 4.676463 mins
> cat("Nr of nodes explored: ", nrOfNodesExplored)
Nr of nodes explored:  117248
> cat("Frontier max length is: ", maxSizeOfFrontier)
Frontier max length is:  107250
>
```

Figure 5: Configuration - 4 rods and 4 disks



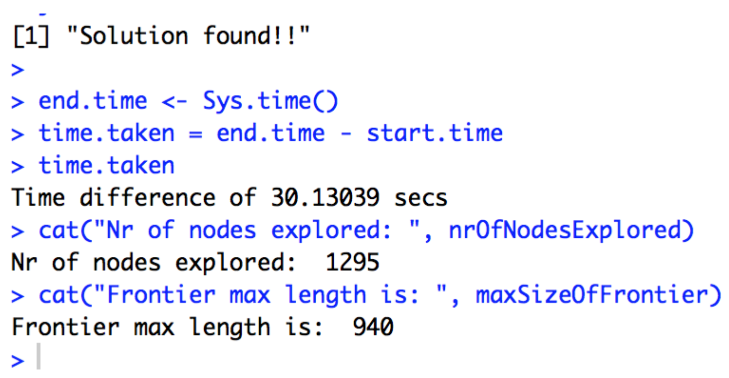
```
[1] "Solution found!!"
>
> end.time <- Sys.time()
> time.taken = end.time - start.time
> time.taken
Time difference of 0.361557 secs
> cat("Nr of nodes explored: ", nrOfNodesExplored)
Nr of nodes explored:  216
> cat("Frontier max length is: ", maxSizeOfFrontier)
Frontier max length is:  138
>
```

Figure 6: Configuration dfs - 6 rods and 4 disks



```
[1] "Solution found!!"
>
> end.time <- Sys.time()
> time.taken = end.time - start.time
> time.taken
Time difference of 2.109473 mins
> cat("Nr of nodes explored: ", nrOfNodesExplored)
Nr of nodes explored:  3908
> cat("Frontier max length is: ", maxSizeOfFrontier)
Frontier max length is:  2245
>
```

Figure 7: Configuration dfs - 4 rods and 6 disks



```
[1] "Solution found!!"
>
> end.time <- Sys.time()
> time.taken = end.time - start.time
> time.taken
Time difference of 30.13039 secs
> cat("Nr of nodes explored: ", nrOfNodesExplored)
Nr of nodes explored:  1295
> cat("Frontier max length is: ", maxSizeOfFrontier)
Frontier max length is:  940
>
```

Figure 8: Configuration dfs - 6 rods and 4 disks

4 Conclusion

We computed and than proved with the code that developing entire tree might take too much computational power. The DFS, on the other hand found solution for all configurations.It was a really interesting assignment and we've learned a lot of things. It was really funny to calculate how many nodes should be expended in order to find solution without using DFS algorithm.