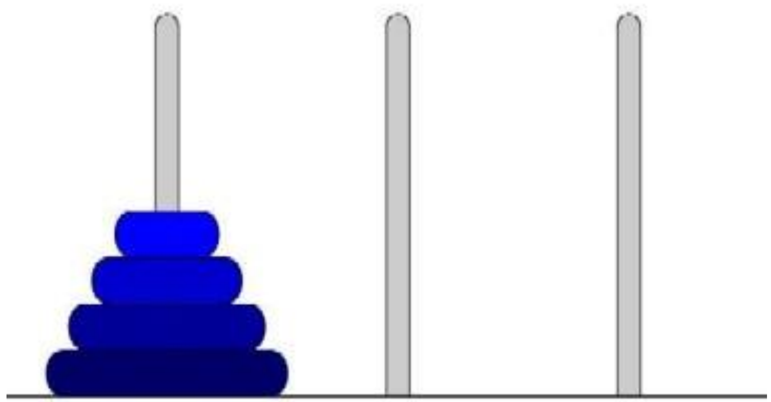
Recursion

Tower of Hanoi

Instructor: Samreen Ishfaq

Tower of Hanoi

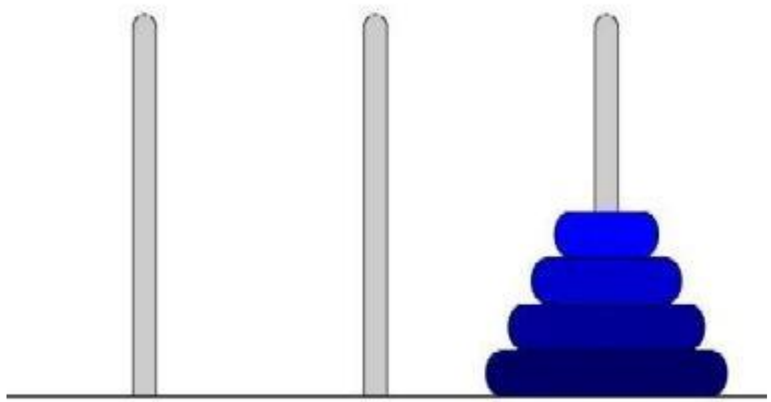
- Tower of Hanoi is a mathematical puzzle.
- The game starts by having few discs stacked in increasing order of size. The number of discs can vary, but there are <u>only</u> three pegs.



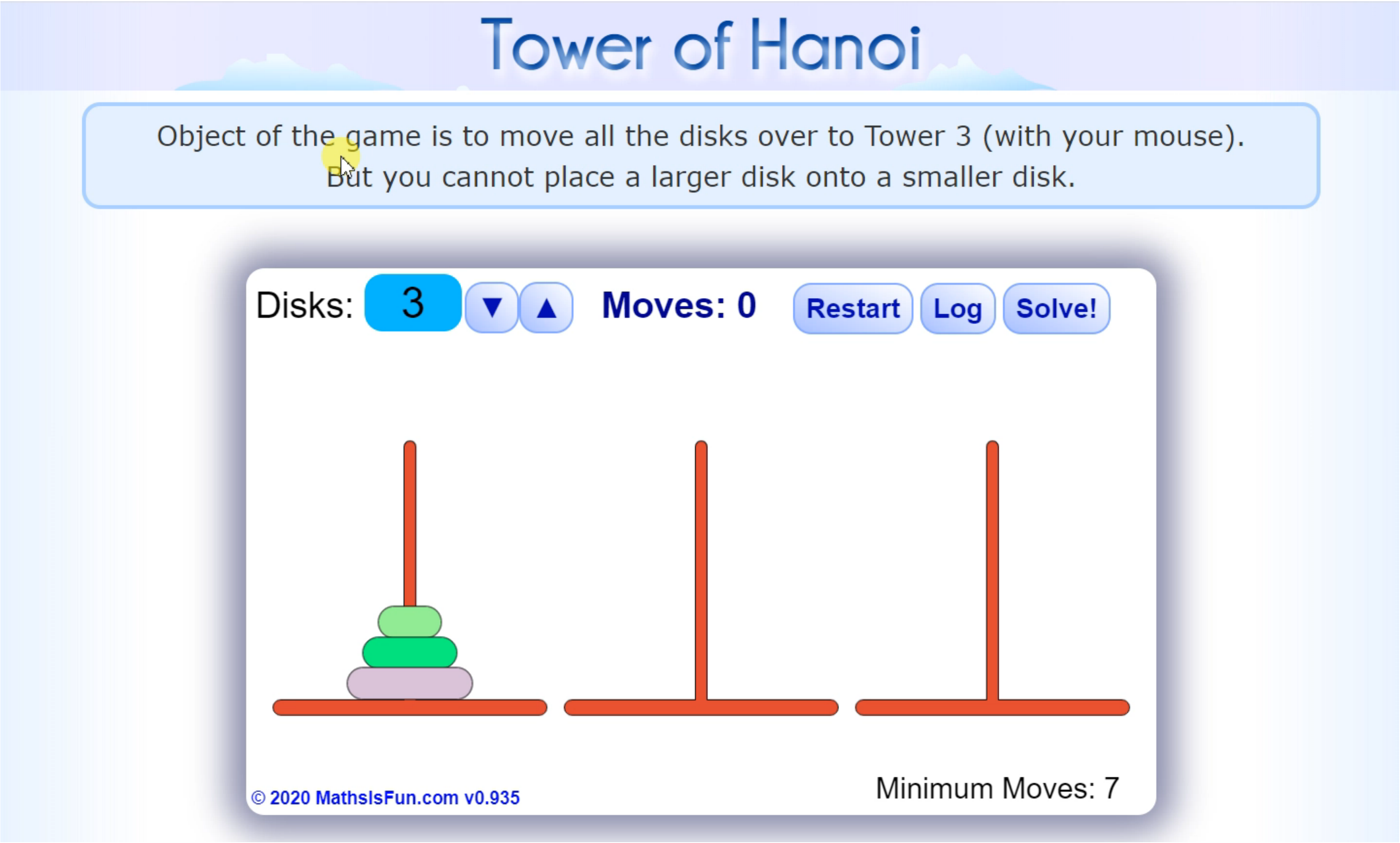
Tower of Hanoi

https://www.mathsisfun.com/games/towerofhanoi.html

- The Objective is to transfer the entire tower to one of the other pegs. However you can only move one disk at a time and you can never stack a larger disk onto a smaller disk. Try to solve it in fewest possible moves.

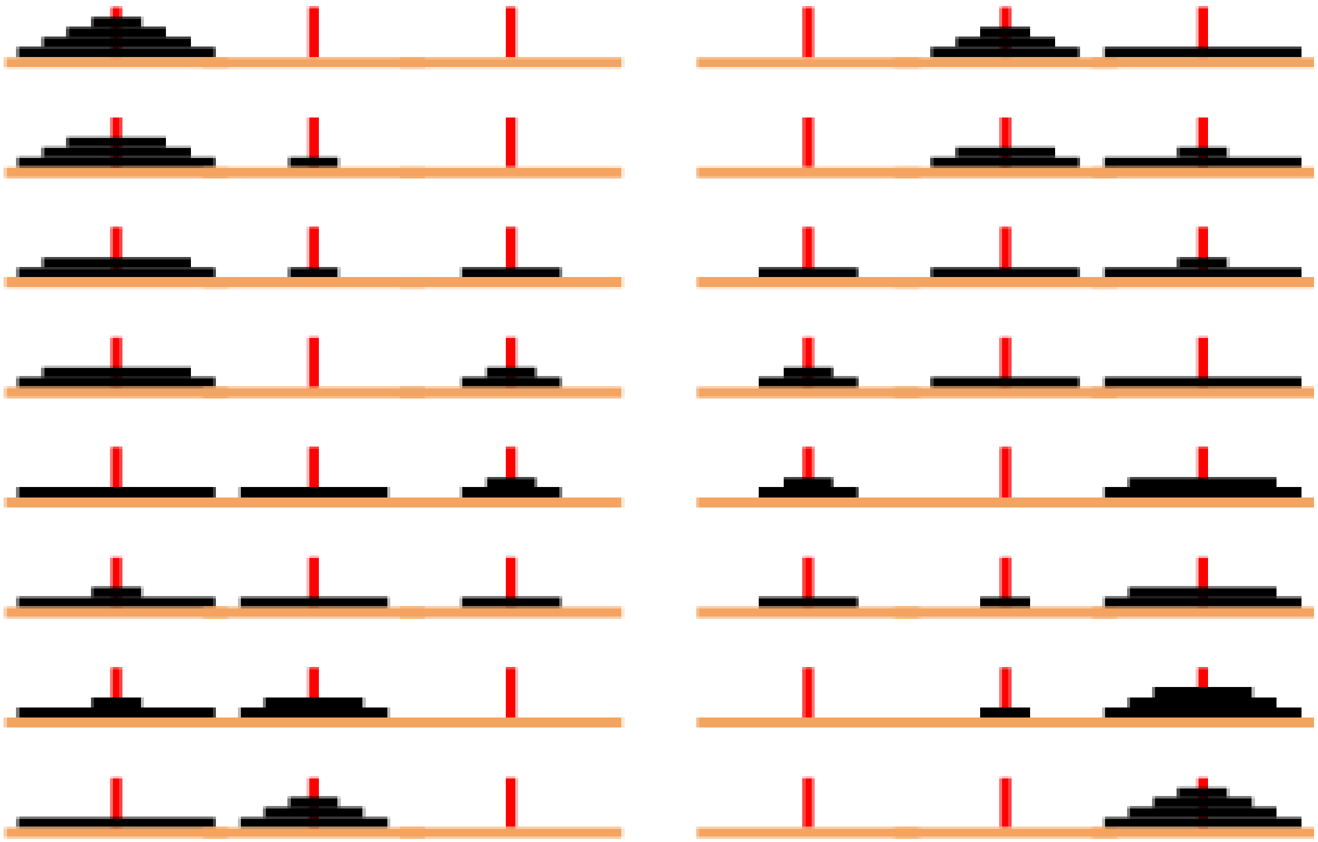


Tower of Hanoi (3 discs)



Tower of Hanoi

How to solve the 4 pegs



Instructor: Samreen Ishfaq

Tower of Hanoi

Solution

To get a better understanding for the general algorithm used to solve the Tower of Hanoi, try to solve the puzzle with a small amount of disks, 3 or 4, and once you master that , you can solve the same puzzle with more discs with the following algorithm.

Tower of Hanoi

Recursive Solution for the Tower of Hanoi with algorithm

Let's call the three peg Src(Source), Spare(temporary) and dst(Destination).

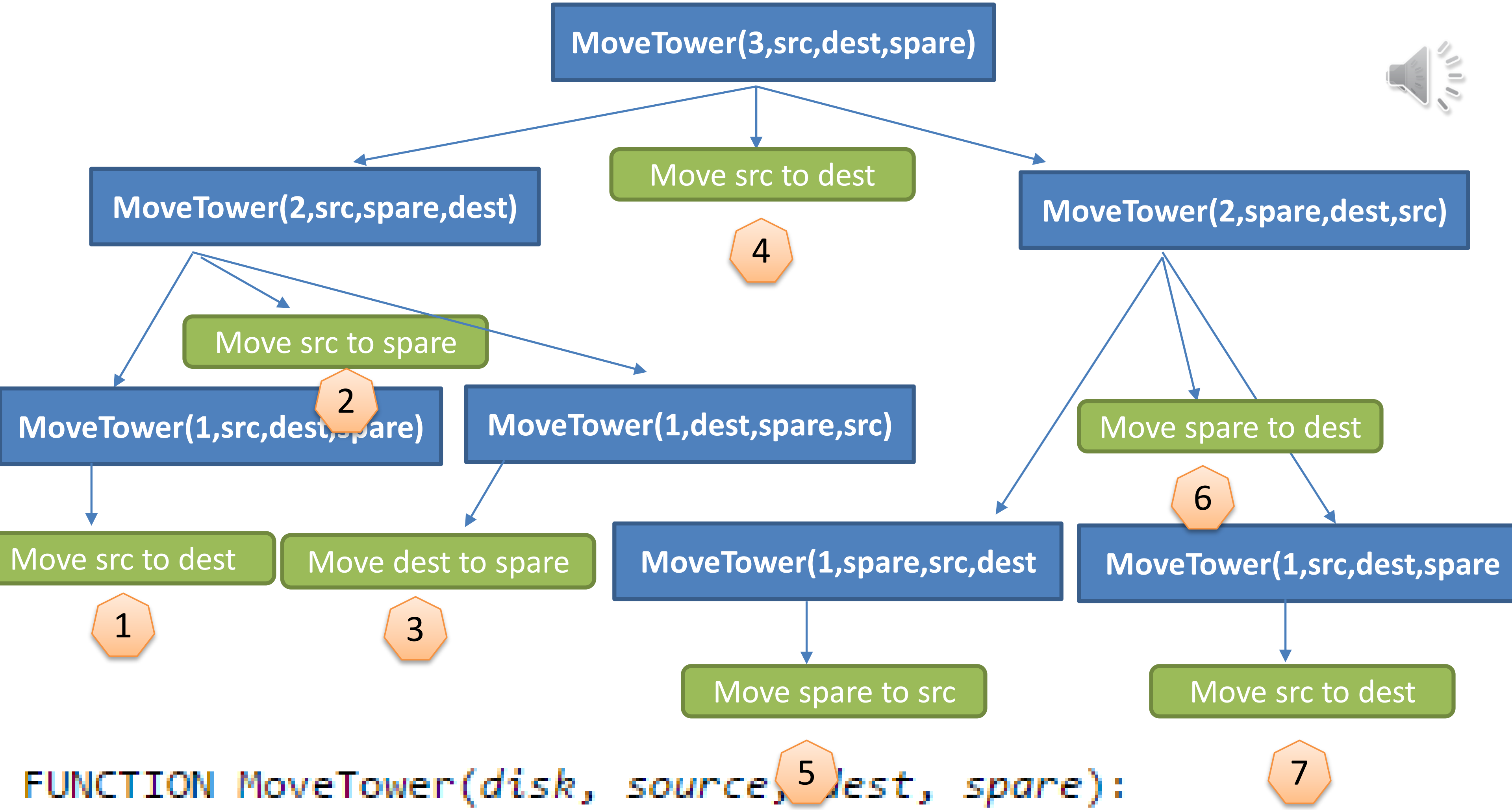
1) Move the top N – 1 disks from the Source to Spare tower
2) Move the Nth disk from Source to Destination tower
3) Move the N – 1 disks from Spare tower to Destination tower.

So once you master solving Tower of Hanoi with three disks, you can solve it with any number of disks with the above algorithm.

Tower of Hanoi

A function solve with four arguments (number of disks) and three pegs (source,spare and destination) could look like this.

```
FUNCTION MoveTower(disk, source, dest, spare):
IF disk == 1, THEN:
    move disk from source to dest
ELSE:
    MoveTower(disk - 1, source, spare, dest)
    move disk from source to dest
    MoveTower(disk - 1, spare, dest, source)
END IF
```




```
FUNCTION MoveTower(disk, source, dest, spare):
IF disk == 0, THEN:
    move disk from source to dest
ELSE:
    MoveTower(disk - 1, source, spare, dest)
    move disk from source to dest
    MoveTower(disk - 1, spare, dest, source)
END IF
```

Tower of Hanoi

To me the sheer simplicity of the solution is breathtaking. For N =3 it translates into

1. Move from Src to Dst
2. Move from Src to Spare
3. Move from Dst to Spare
4. Move from Src to Dst
5. Move from Spare to Src
6. Move from Spare to Dst
7. Move from Src to Dst

Of course "Move" means moving the topmost disk.

Tower of Hanoi

For N = 4 we get the following sequence

1. Move from Src to Spare
2. Move from Src to Dst
3. Move from Spare to Dst
4. Move from Src to Spare
5. Move from Dst to Src
6. Move from Dst to Spare
7. Move from Src to Spare
8. Move from Src to Dst
9. Move from Spare to Dst
10. Move from Spare to Src
11. Move from Dst to Src
12. Move from Spare to Dst
13. Move from Src to Spare
14. Move from Src to Dst
15. Move from Spare to Dst

Tower of Hanoi

How many moves will it take to transfer n disks from the left post to the right post?

- the recursive pattern *can* help us generate more numbers to find an *explicit* (non-recursive) pattern. Here's how to find the number of moves needed to transfer larger numbers of disks from post A to post C, remembering that M = the number of moves needed to transfer n-1 disks from post A to post C:
- for **1 disk** it takes 1 move to transfer 1 disk from post A to post C;
- for **2 disks**, it will take 3 moves: $2M + 1 = 2(\mathbf{1}) + 1 = \mathbf{3}$
- for **3 disks**, it will take 7 moves: $2M + 1 = 2(\mathbf{3}) + 1 = \mathbf{7}$
- for **4 disks**, it will take 15 moves: $2M + 1 = 2(\mathbf{7}) + 1 = \mathbf{15}$
- for **5 disks**, it will take 31 moves: $2M + 1 = 2(\mathbf{15}) + 1 = \mathbf{31}$
- for **6 disks**... ?

Tower of Hanoi

- **Explicit Pattern**
- Number of Disks Number of Moves

Number of Disks	Number of Moves
1	1
2	3
3	7
4	15
5	31

2^n-1

- *Powers of two help reveal the pattern:*
- Number of Disks (n) Number of Moves

Number of Disks (n)	Number of Moves
1	2^1 - 1 = 2 - 1 = 1
2	2^2 - 1 = 4 - 1 = 3
3	2^3 - 1 = 8 - 1 = 7
4	2^4 - 1 = 16 - 1 = 15
5	2^5 - 1 = 32 - 1 = 31