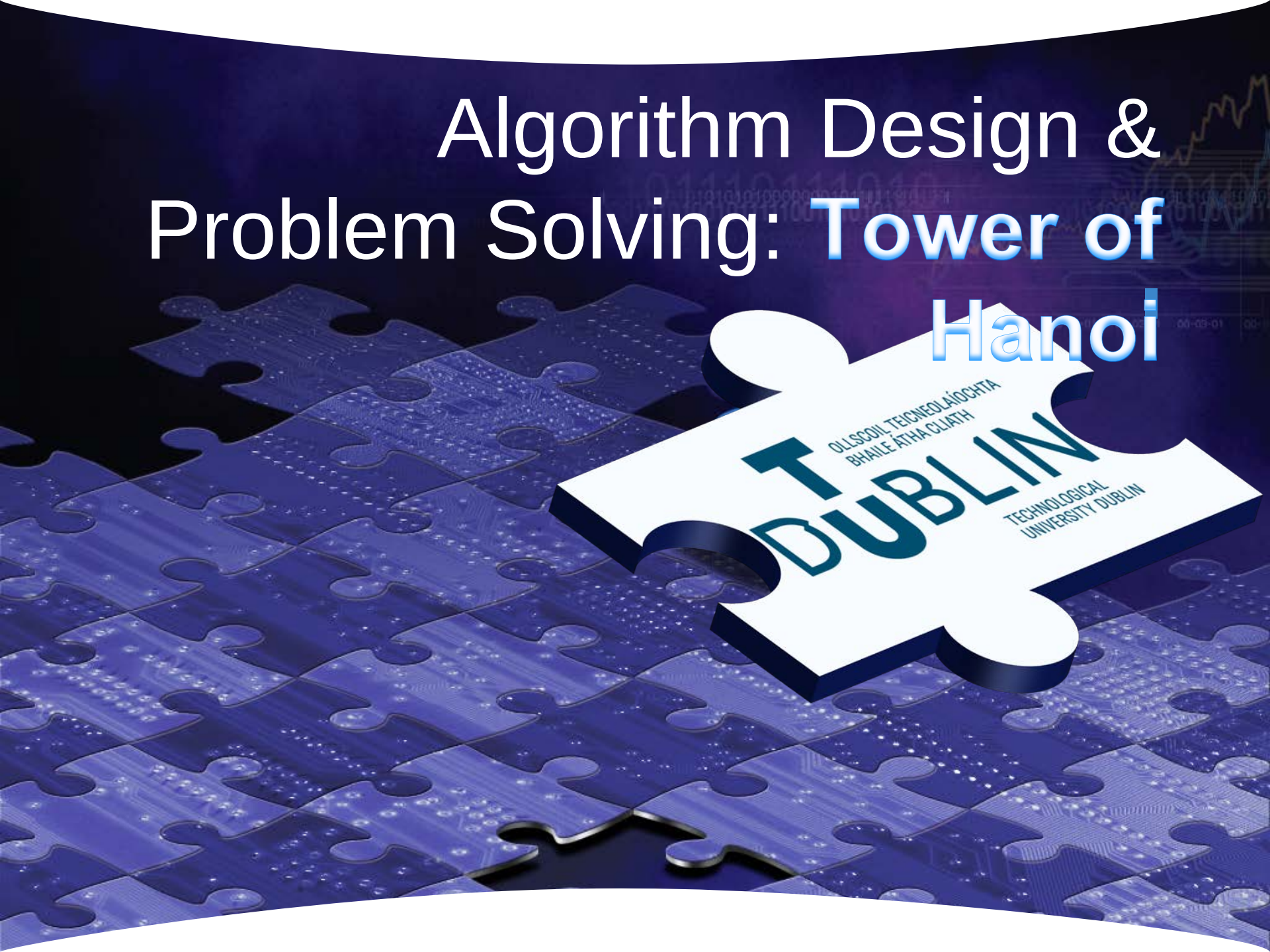
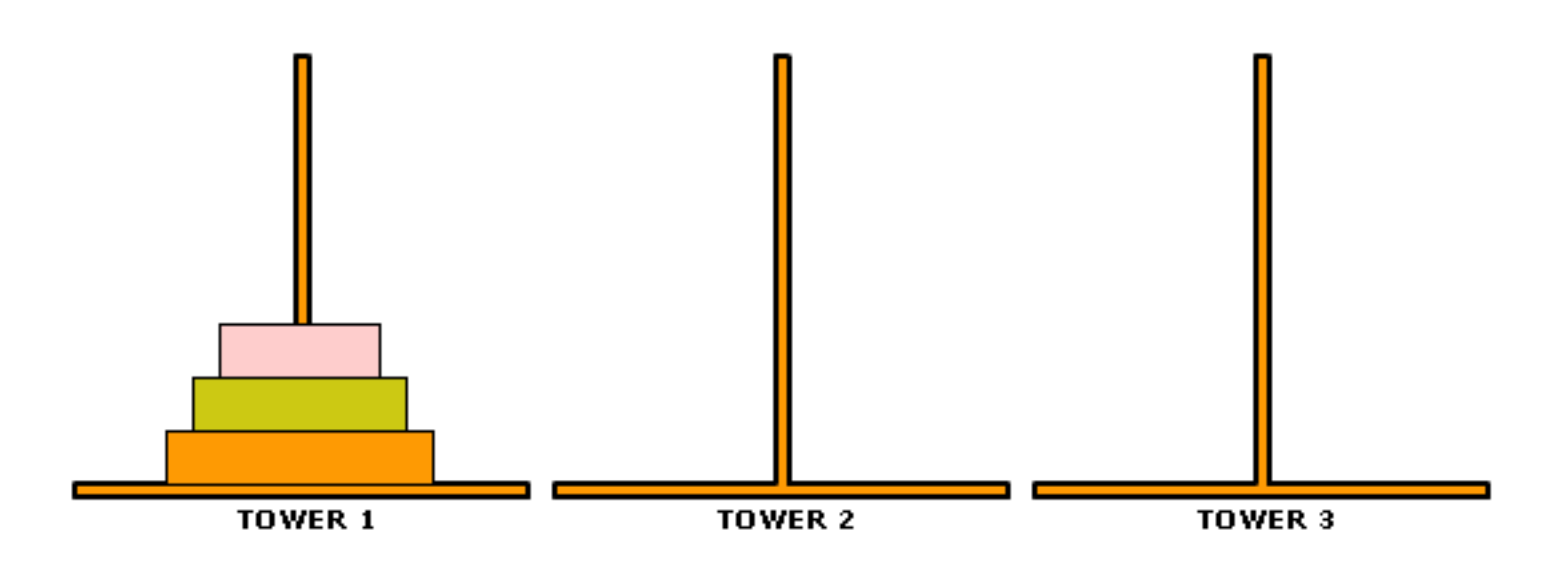
Algorithm Design & Problem Solving: Tower of Hanoi

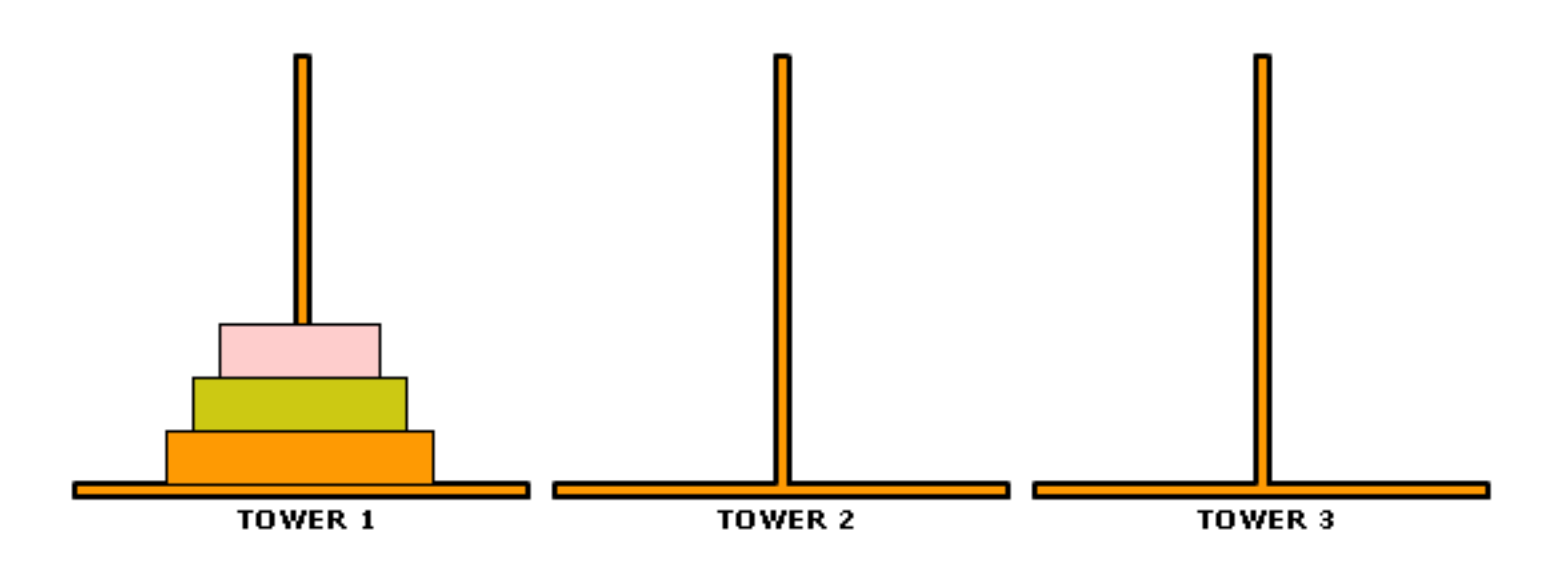


The Tower of Hanoi



- **Rules**
  - Move all disks to Tower 3
  - Only one disk can be moved at a time
  - A disk can never be put on a smaller disk

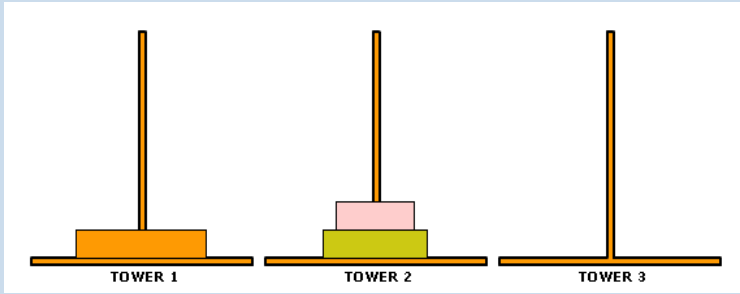
The Tower of Hanoi



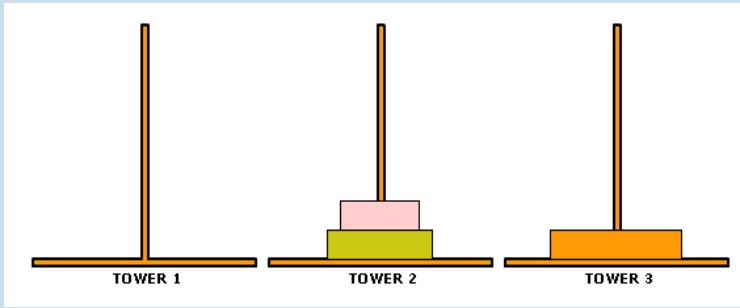
- What is the problem?
  - Move the largest disk, disk2, to Tower 3
  - Move the middle disk, disk1, to Tower 3
  - Move smallest disk, disk0, to Tower 3

The Tower of Hanoi

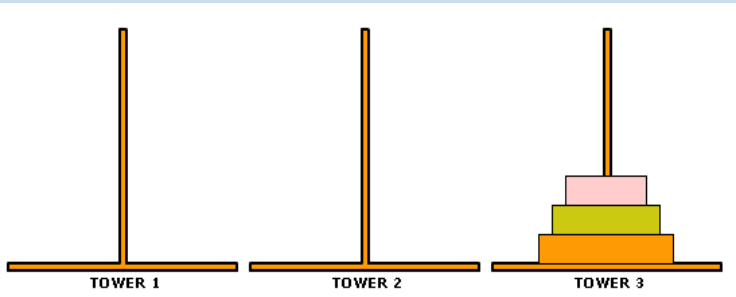
- To move disk0, disk1 and disk2 from 1 to 3:
  1. Move disk0 and disk1 from 1 to 2,
     i. Move disk0 from 1 to 3, and
     ii. Move disk1 from 1 to 2, and
     iii. Move disk0 from 3 to 2



  2. Move disk2 from 1 to 3



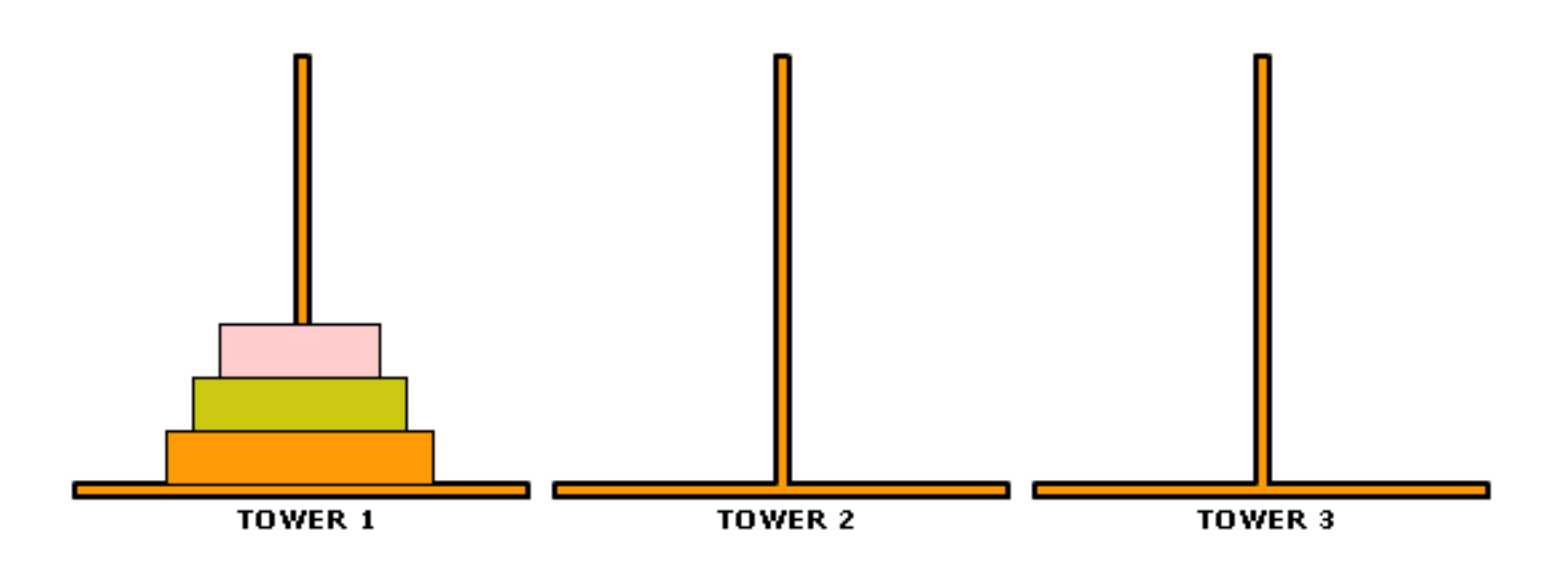
  3. Move disk0 and disk1 from 2 to 3
     i. Move disk0 from 2 to 1, and
     ii. Move disk1 from 2 to 3, and
     iii. Move disk0 from 1 to 3



Towers of Hanoi

- Why is this suitable for **recursion**?
  - Because there is a base case
  - The problem is iteratively getting smaller

The Tower of Hanoi



Source

Spare

Dest

Towers of Hanoi: A recursive algorithm

Which disk am I moving?

moveTower (disks, source, dest, spare)

From which tower?

To which tower?

In your TOH handout from yesterday, fill up the column “Relevant function call”

Towers of Hanoi: A recursive algorithm

Tower of Hanoi

Actual Moves	Line of code that move takes place	Relevant function call
Move ___ from Tower ___ to Tower ___		moveTower(0,T1,T3,T2)
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		

Towers of Hanoi: A recursive algorithm….. Start of

moveTower (disks, source, dest, spare)

```
If disk = 0
        Move disk from source to dest
```

Base case: if it is the smallest disk then move it

Towers of Hanoi: A recursive algorithm

moveTower (disks, source, dest, spare)

If disk = 0

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)

Base case: only small disk can be moved

Towers of Hanoi: A recursive algorithm

```
moveTower (disks, source, dest, spare)
        If disk = 0
                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)
```

This moves the big & medium disks

Towers of Hanoi: A recursive algorithm

moveTower (disks, source, dest, spare)

If disk = 0

Move disk from source to dest

else

moveTower (disk-1, source, spare, dest)

Recursive calls

move disk from source to dest

moveTower (disk-1, spare, dest, source)

Towers of Hanoi: A recursive algorithm

```
moveTower (disks, source, dest, spare)
        If disk = 0
                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)
```

Note how the towers change position in the call

Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

In your TOH handout, fill up the column “Line of code that move takes place”

Towers of Hanoi: A recursive algorithm

Tower of Hanoi

Actual Moves	Line of code that move takes place	Relevant function call
Move ___ from Tower ___ to Tower ___	3	moveTower(0,T1,T3,T2)
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		
Move ___ from Tower ___ to Tower ___		

Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

moveTower(2,Tower1,Tower3,Tower2)

Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

moveTower(2,Tower1,Tower3,Tower2)

TOP OF STACK

Towers of Hanoi: A recursive algorithm

The call was made at line 5

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

5

Call Stack

moveTower(2,Tower1,Tower3,Tower2)
moveTower(1,Tower1,Tower2,Tower3)

Towers of Hanoi: A recursive algorithm

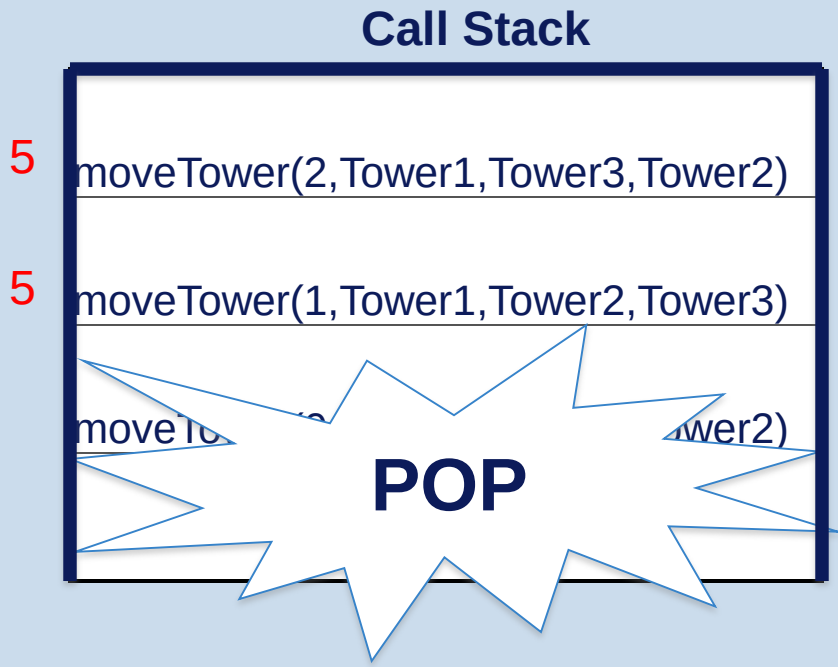
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

5	moveTower(2,Tower1,Tower3,Tower2)
5	moveTower(1,Tower1,Tower2,Tower3)
	moveTower(0,Tower1,Tower3,Tower2)

Towers of Hanoi: A recursive algorithm

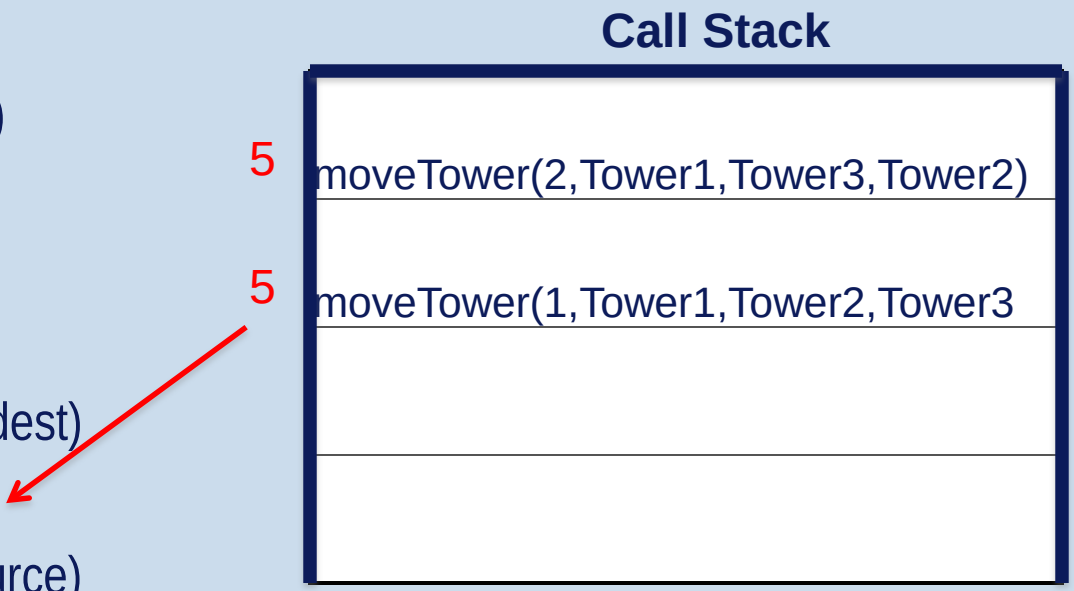
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



1(i) Move disk0 from 1 to 3

Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



1(ii) Move disk1 from 1 to 2

Towers of Hanoi: A recursive algorithm

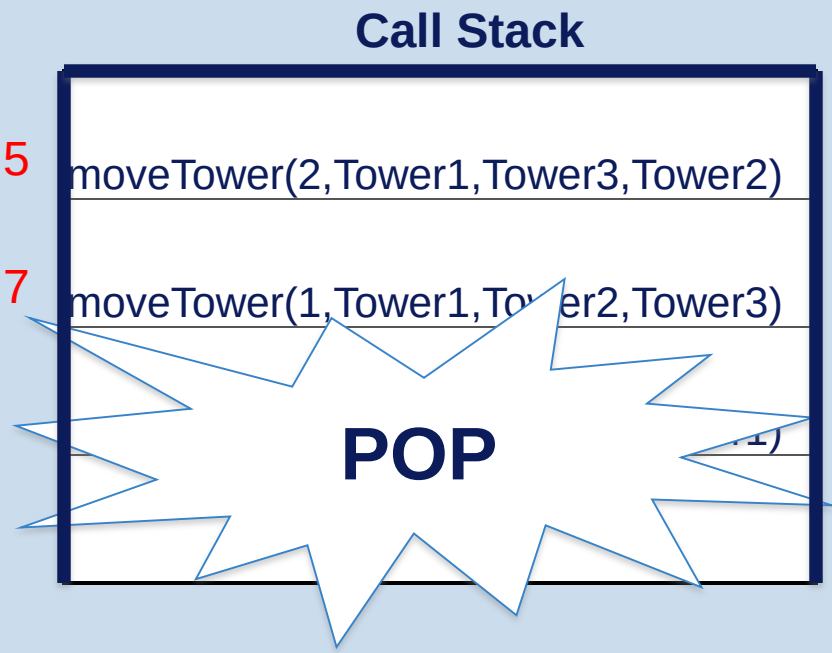
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

5	moveTower(2,Tower1,Tower3,Tower2)
7	moveTower(1,Tower1,Tower2,Tower3)
	moveTower(0,Tower3,Tower2,Tower1)

Towers of Hanoi: A recursive algorithm

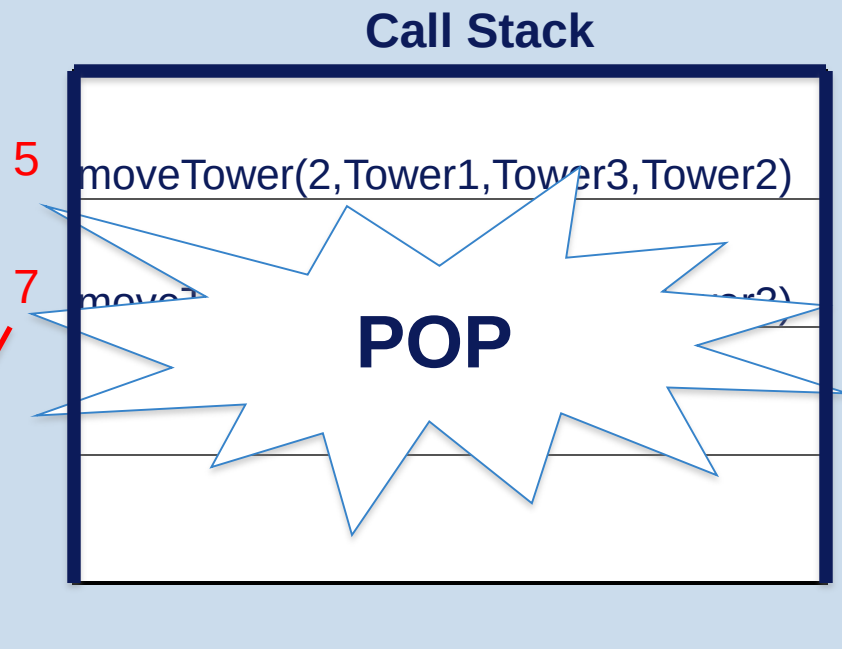
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



1(iii) Move disk0 from 3 to 2

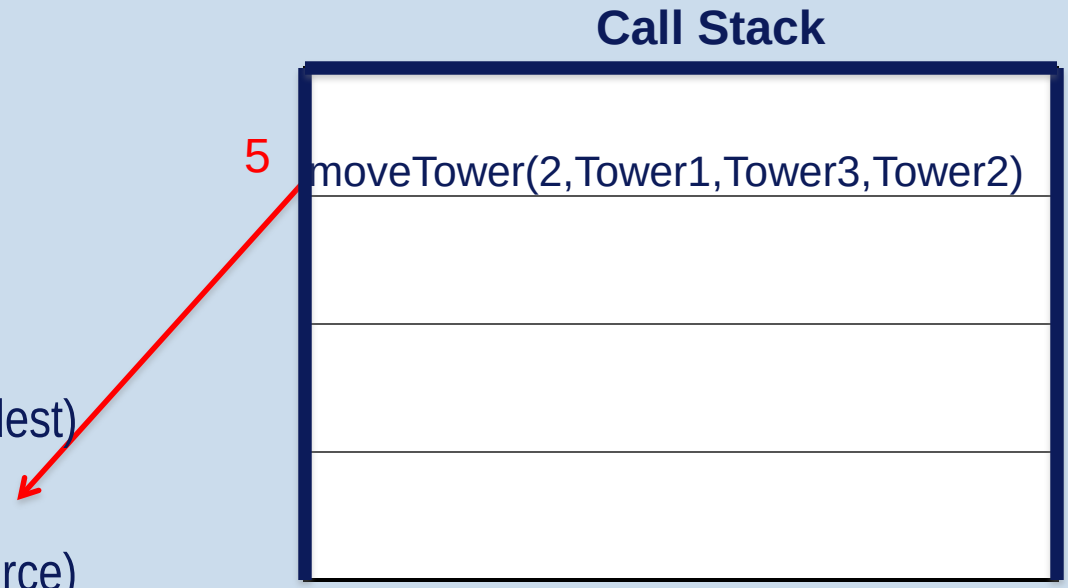
Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



2. Move disk2 from 1 to 3

Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

7

moveTower(2,Tower1,Tower3,Tower2)
moveTower(1,Tower2,Tower3,Tower1)

Towers of Hanoi: A recursive algorithm

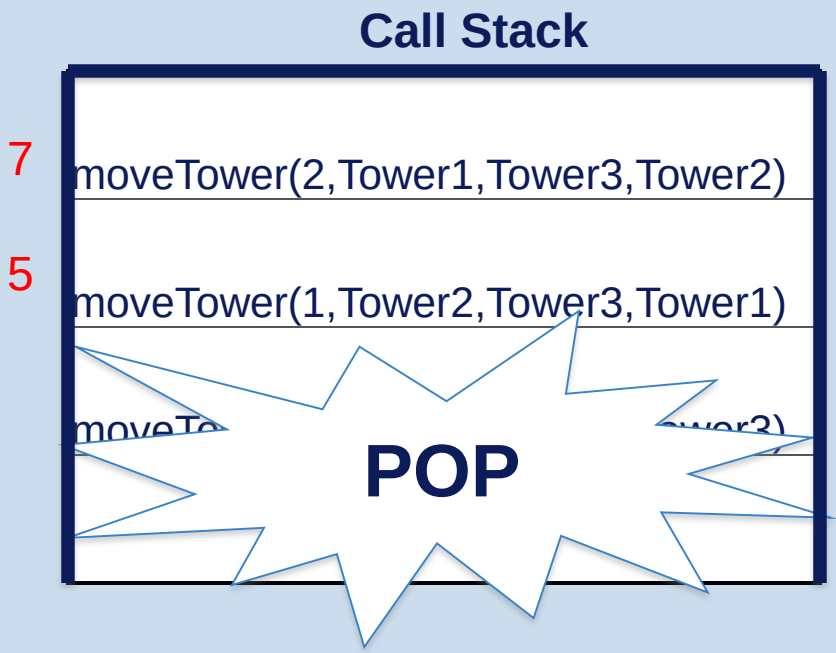
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

7	moveTower(2,Tower1,Tower3,Tower2)
5	moveTower(1,Tower2,Tower3,Tower1)
	moveTower(0,Tower2,Tower1,Tower3)

Towers of Hanoi: A recursive algorithm

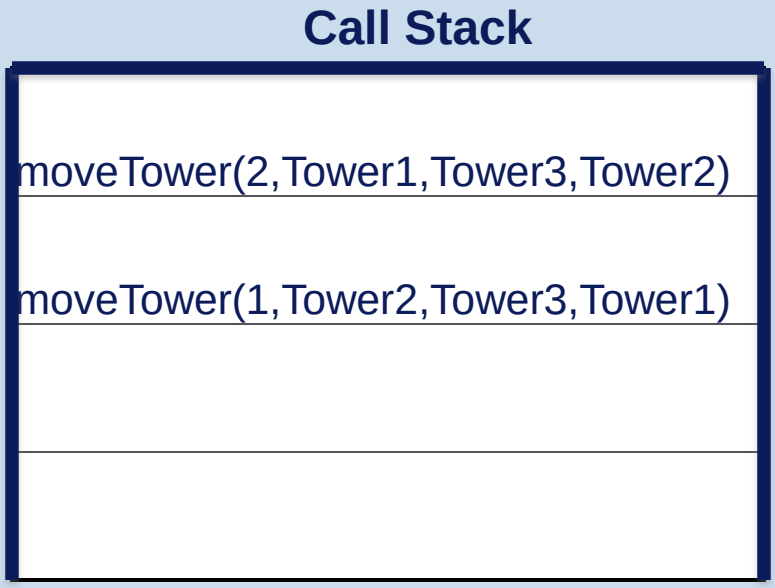
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



3(i) Move disk0 from 2 to 1

Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



3(ii) Move disk1 from 2 to 3

Towers of Hanoi: A recursive algorithm

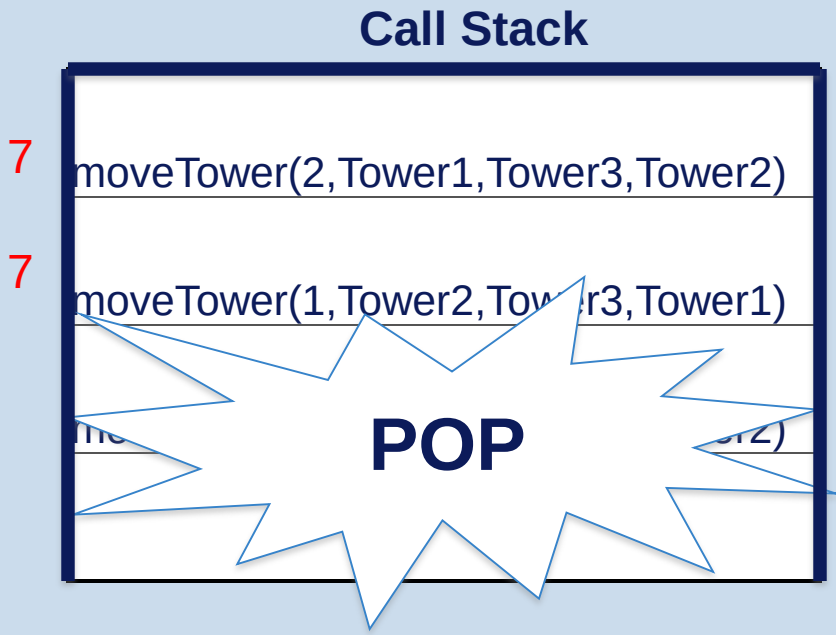
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

7	moveTower(2,Tower1,Tower3,Tower2)
7	moveTower(1,Tower2,Tower3,Tower1)
	moveTower(0,Tower1,Tower3,Tower2)

Towers of Hanoi: A recursive algorithm

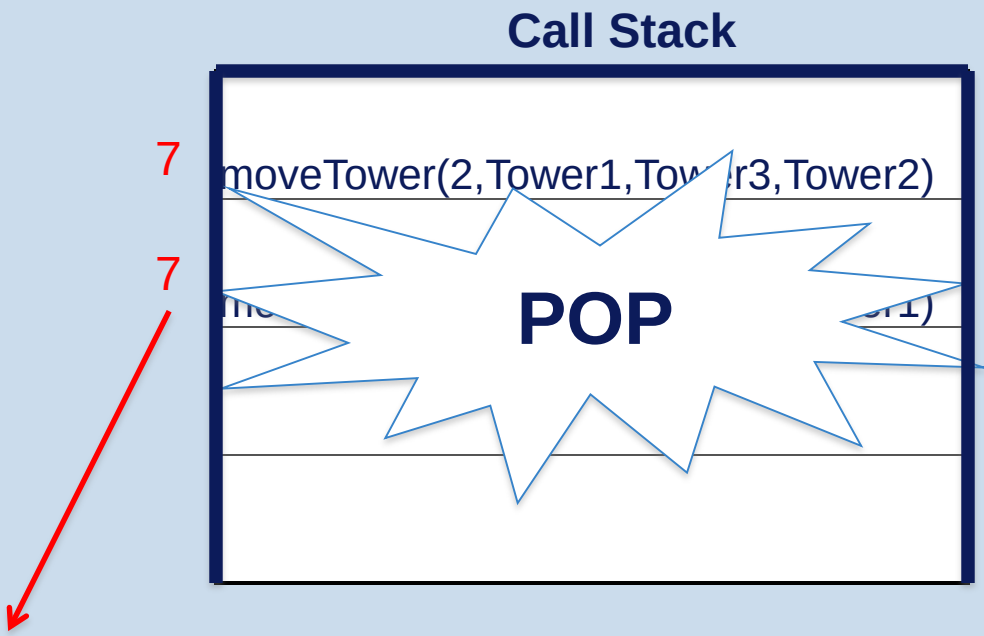
```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



3(iii) Move disk0 from 1 to 3

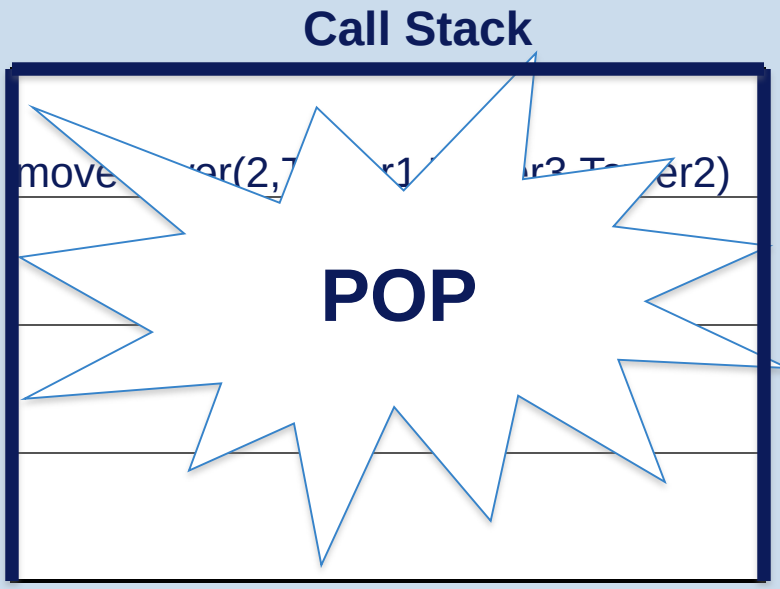
Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```



Towers of Hanoi: A recursive algorithm

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

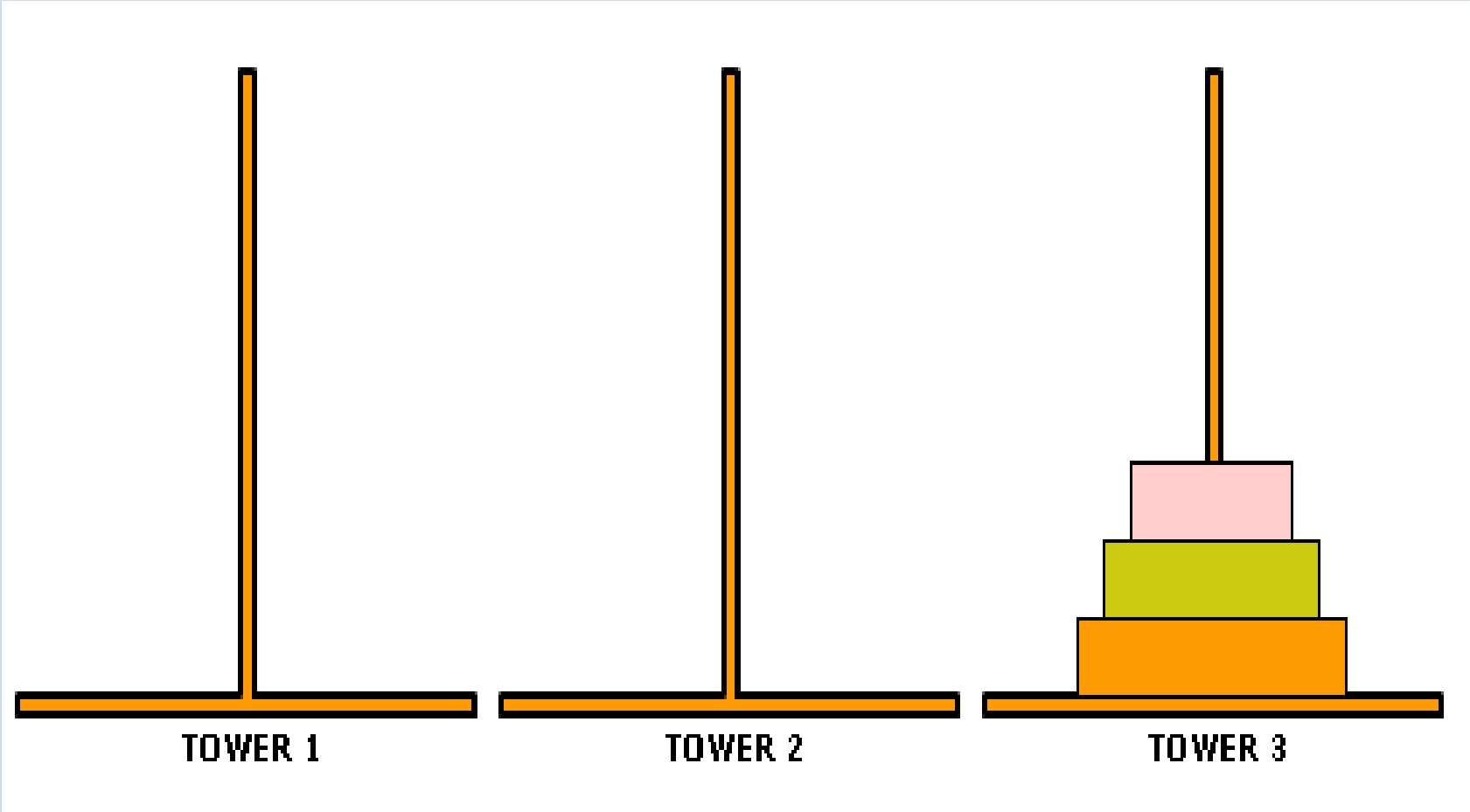


Towers of Hanoi: Bingo!!!

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest)
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Call Stack

Towers of Hanoi: Bingo!!!



Thank You !


OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
TU
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN