CS302 - Data Structures *using C++*

Topic: The Towers of Hanoi

Kostas Alexis

AUTONOMOUS ROBOTS LAB

The Towers of Hanoi

- **Introduction to the problem**
  - The “Towers of Hanoi” is a mathematical puzzle where one has three pegs and n disks and the goal is to move the entire stack to another rod, obeying a set of rules.
- **The Rules of the “Towers of Hanoi”**
  - Only one disk may be moved at a time
  - Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack: a disk can only be moved if it is the uppermost disk on a stack.
  - No disk may be placed on top of a smaller disk.

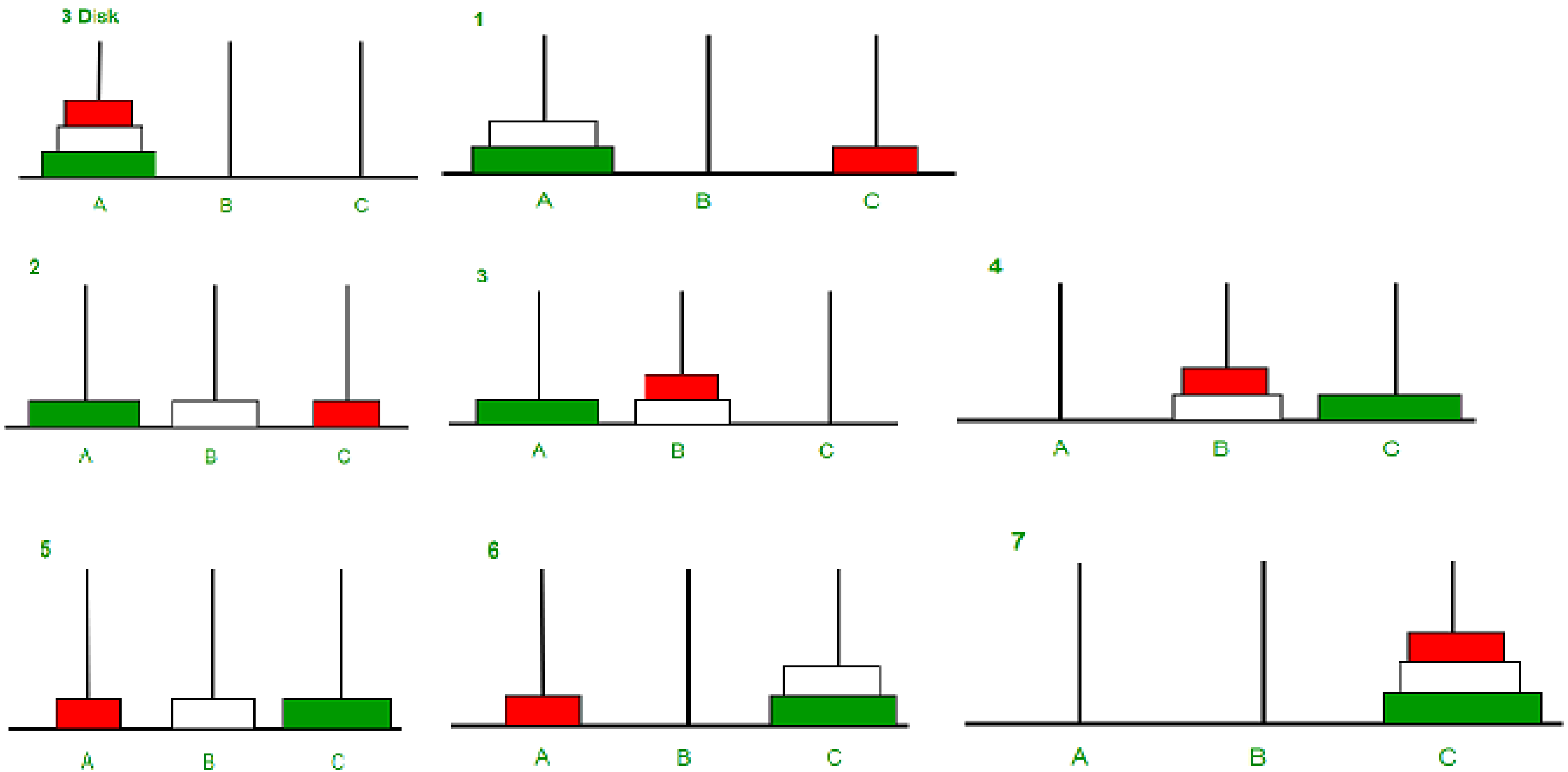
The Towers of Hanoi

- **Problem Statement**
  - Beginning with n disks on pole A and zero disks on poles B and C, solve `towers(n,A,B,C)`

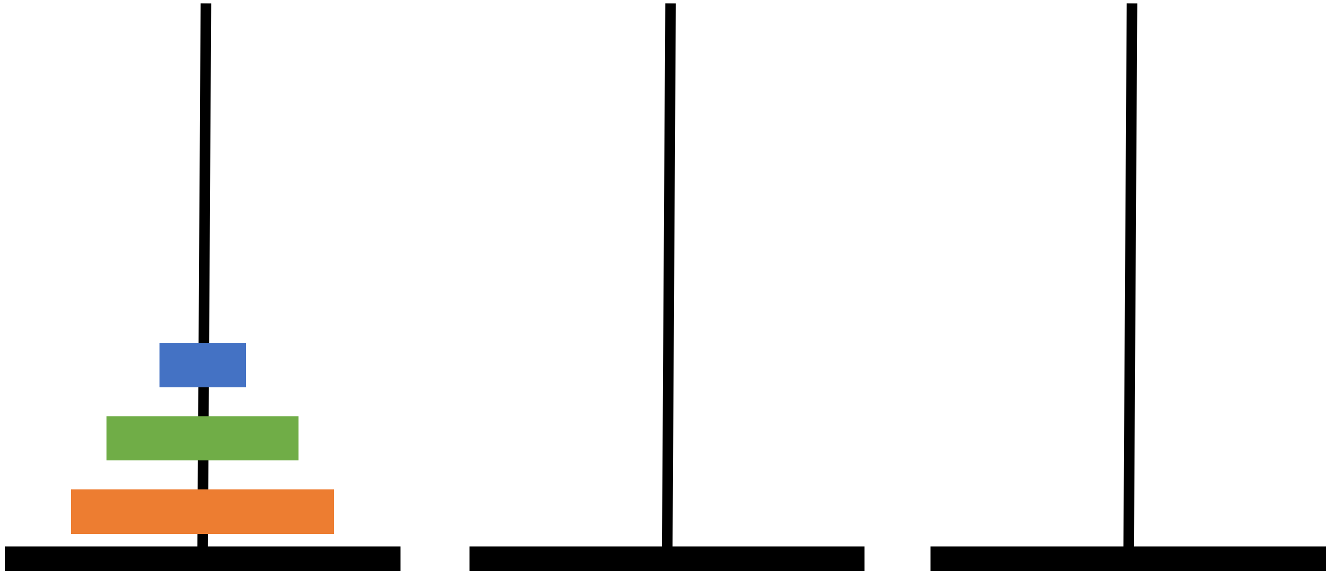
The Towers of Hanoi

- **Problem Statement**
  - Beginning with n disks on pole A and zero disks on poles B and C, solve `towers(n,A,B,C)`
- **Solution**
  - With all disks on A, solve `towers(n-1,A,C,B)`
  - With the largest disk on pole A and all the others on pole C, solve `towers(n-1,A,B,C)`
  - With the largest disk on pole B and all the other disks on pole C, solve `towers(n-1,C,B,A)`

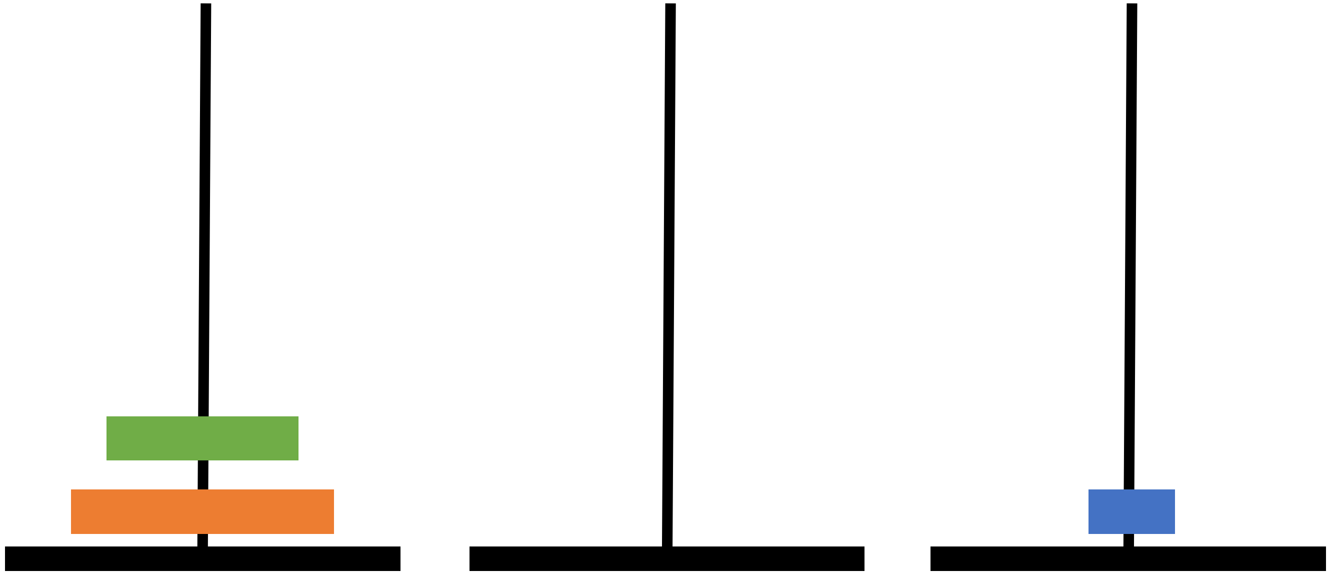
The Towers of Hanoi



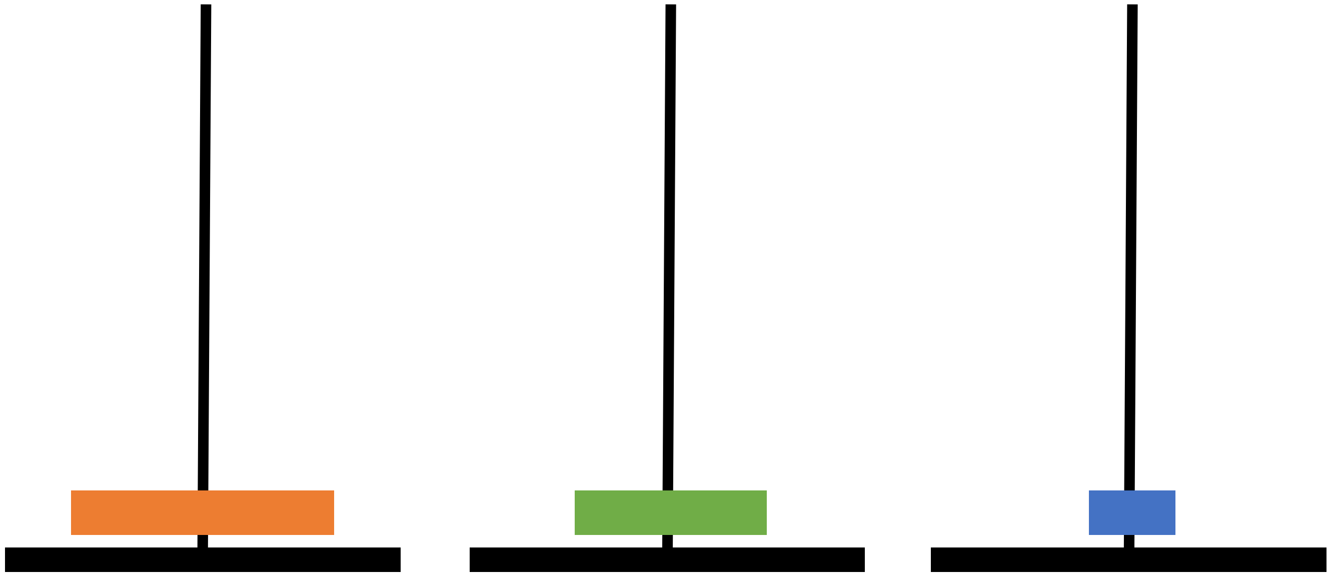
The Towers of Hanoi



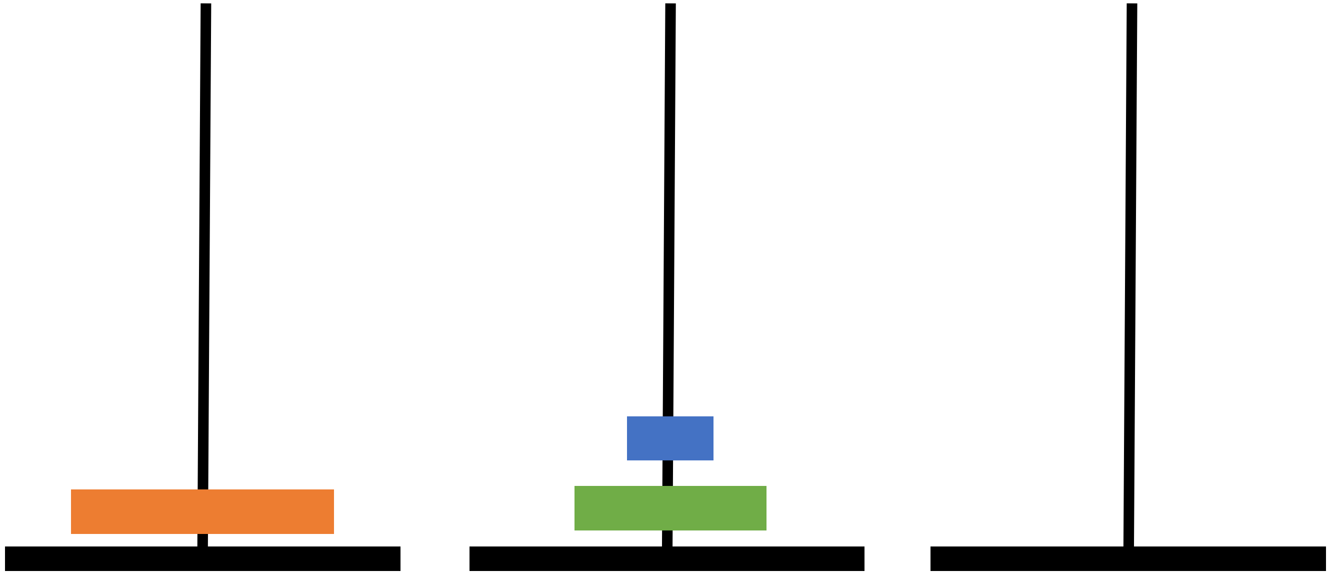
The Towers of Hanoi



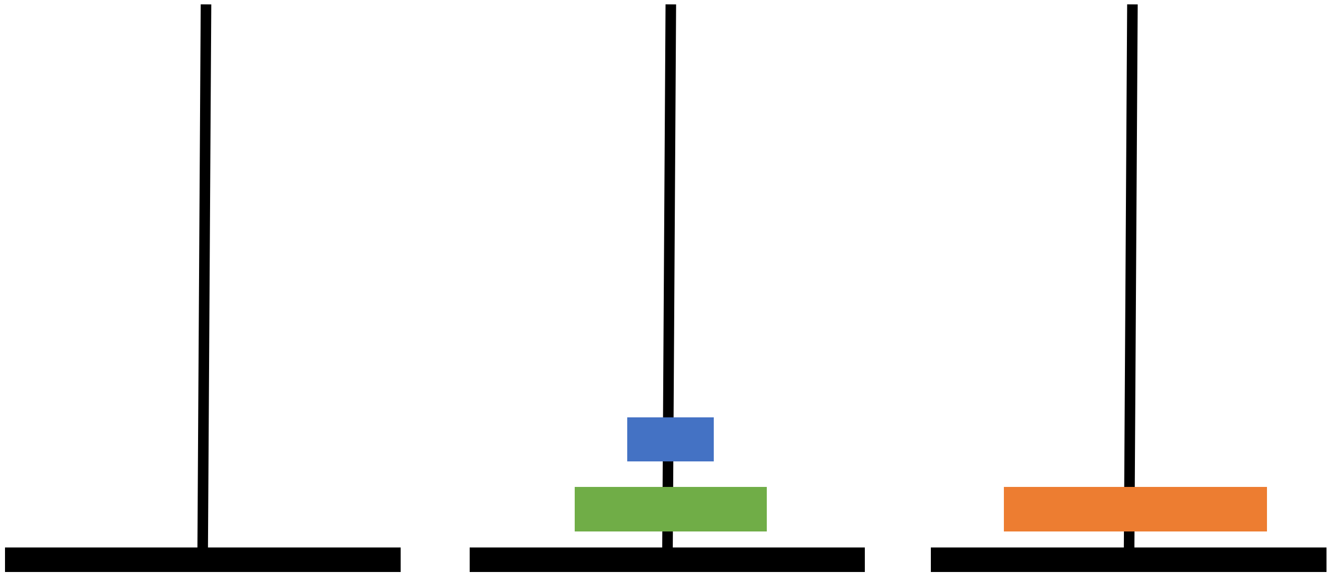
The Towers of Hanoi



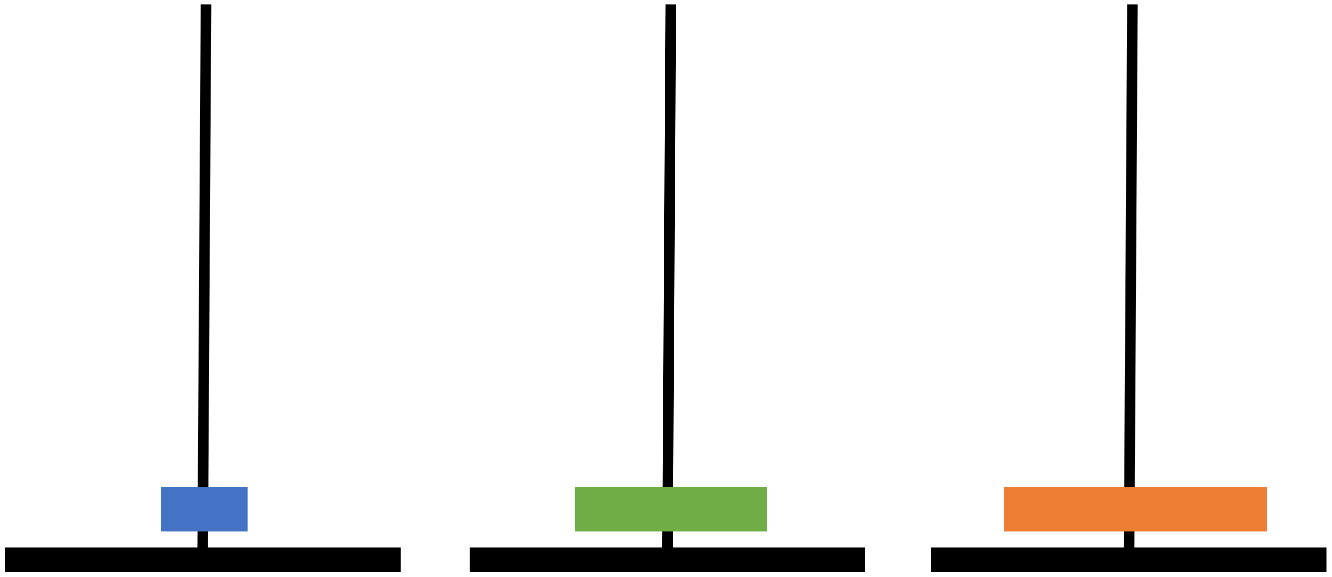
The Towers of Hanoi



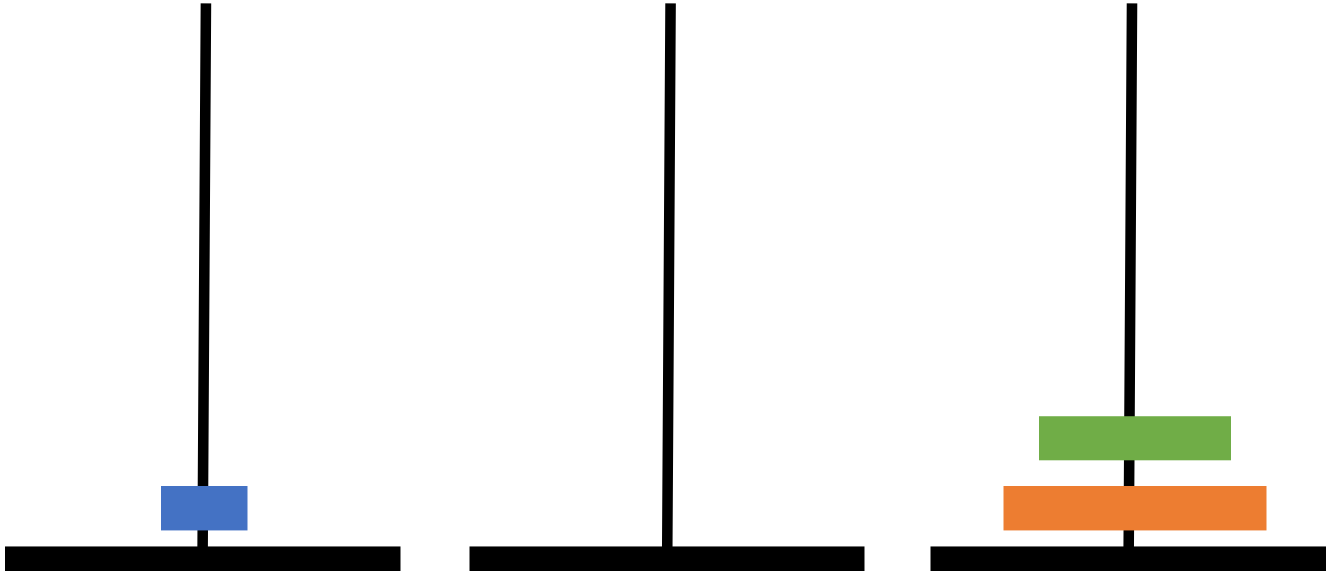
The Towers of Hanoi



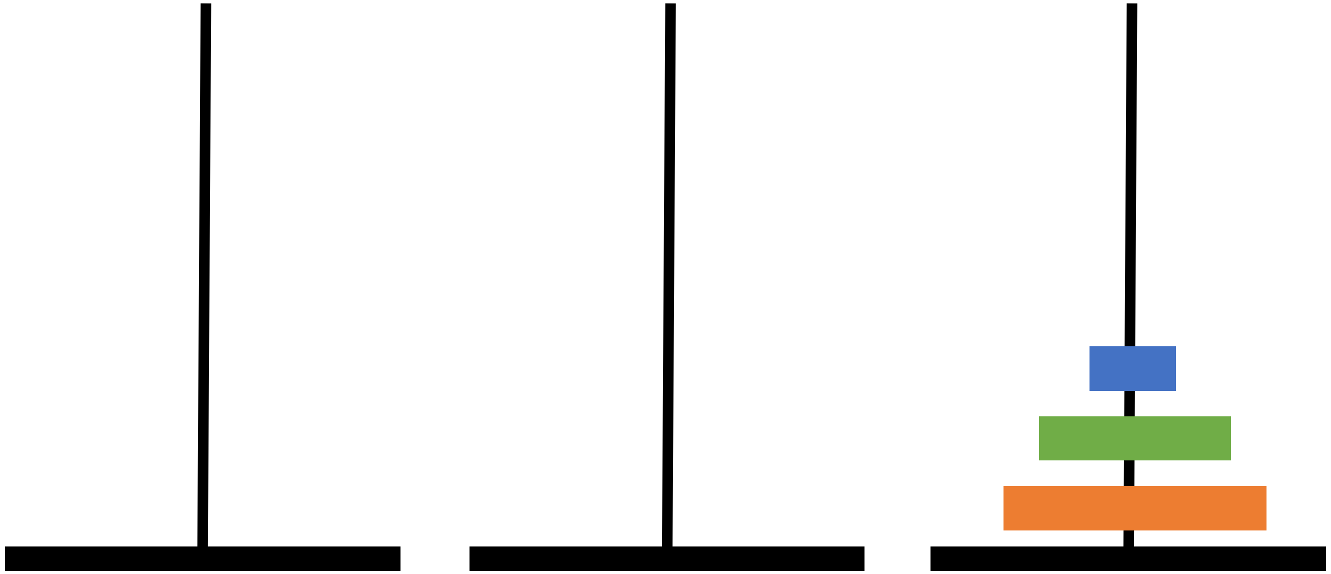
The Towers of Hanoi



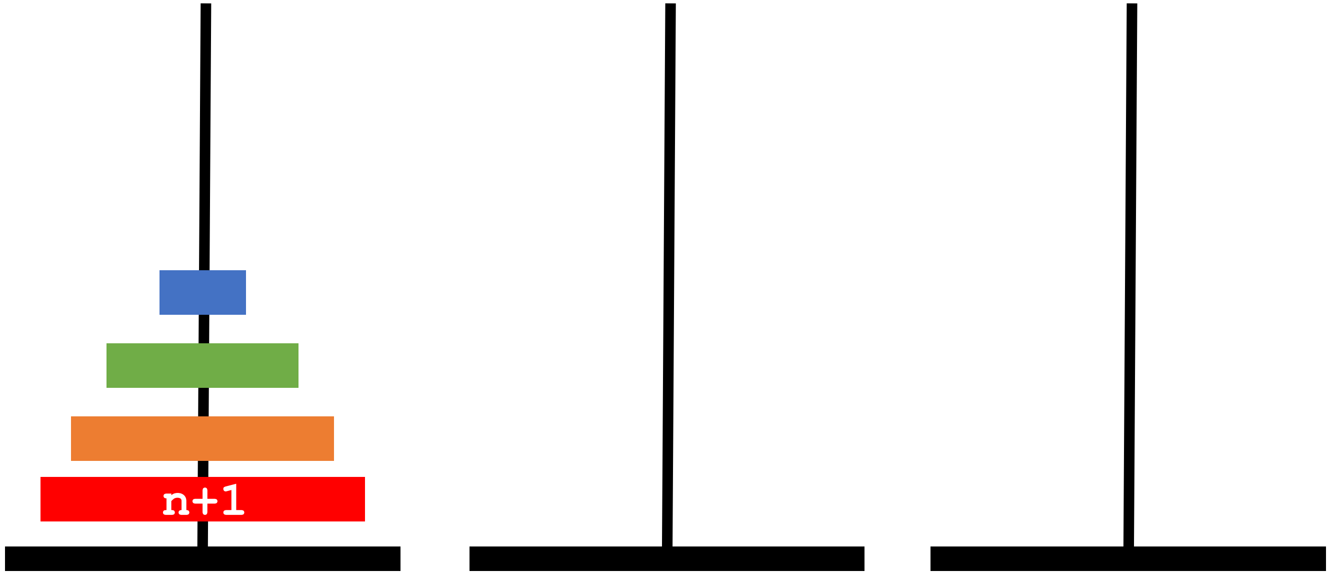
The Towers of Hanoi



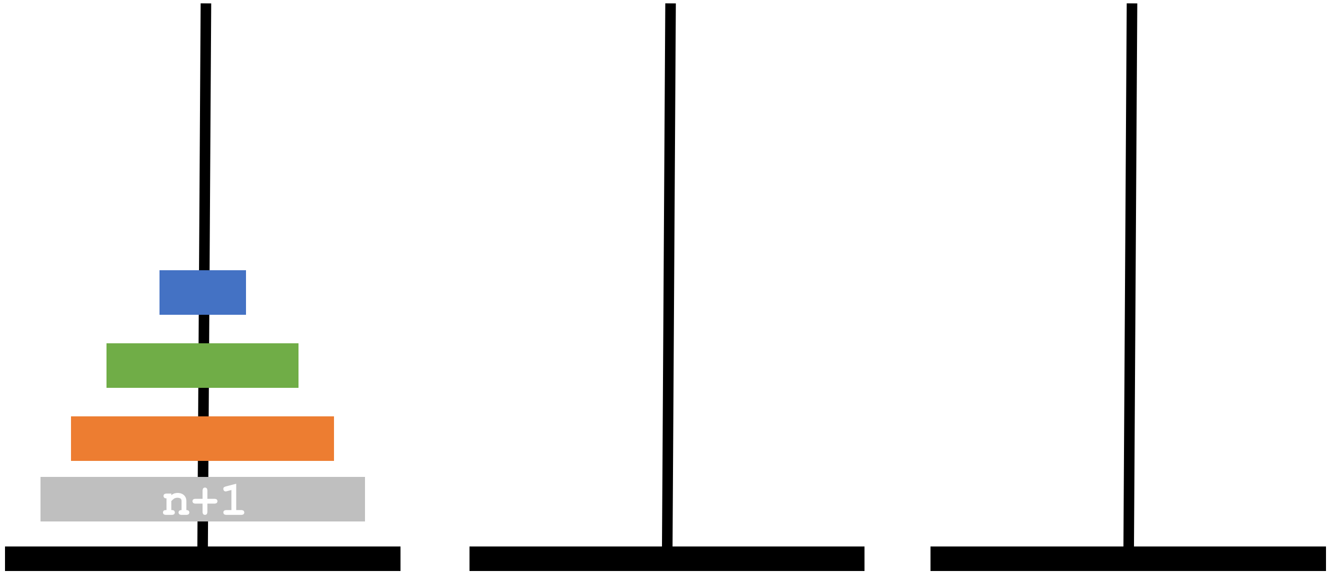
The Towers of Hanoi



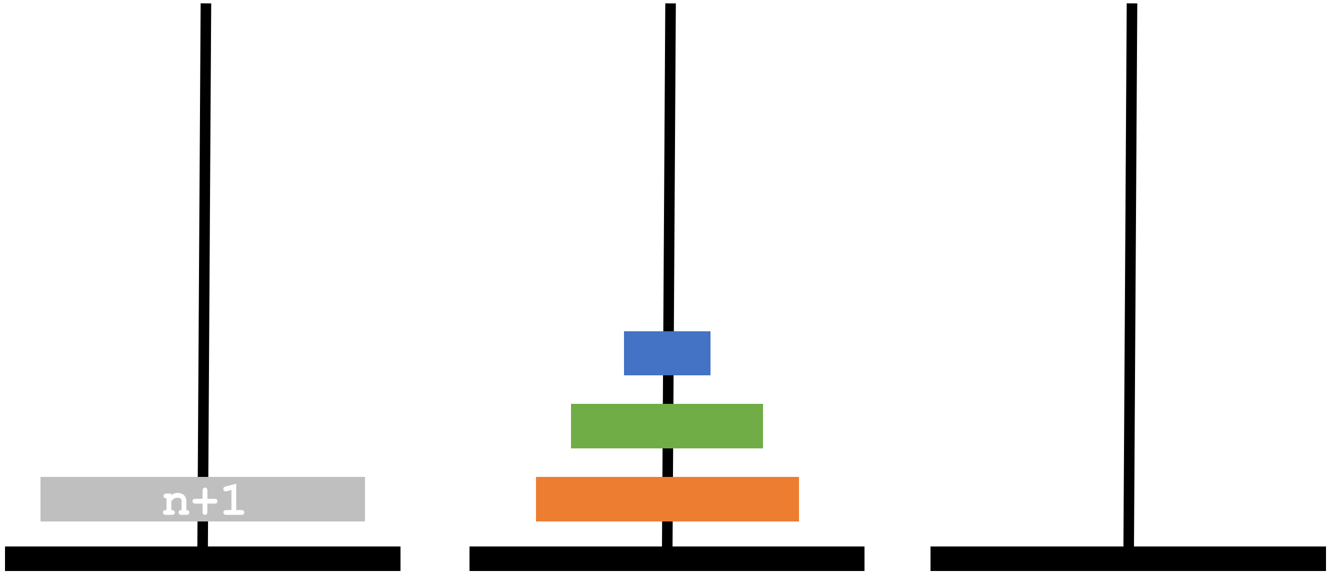
The Towers of Hanoi



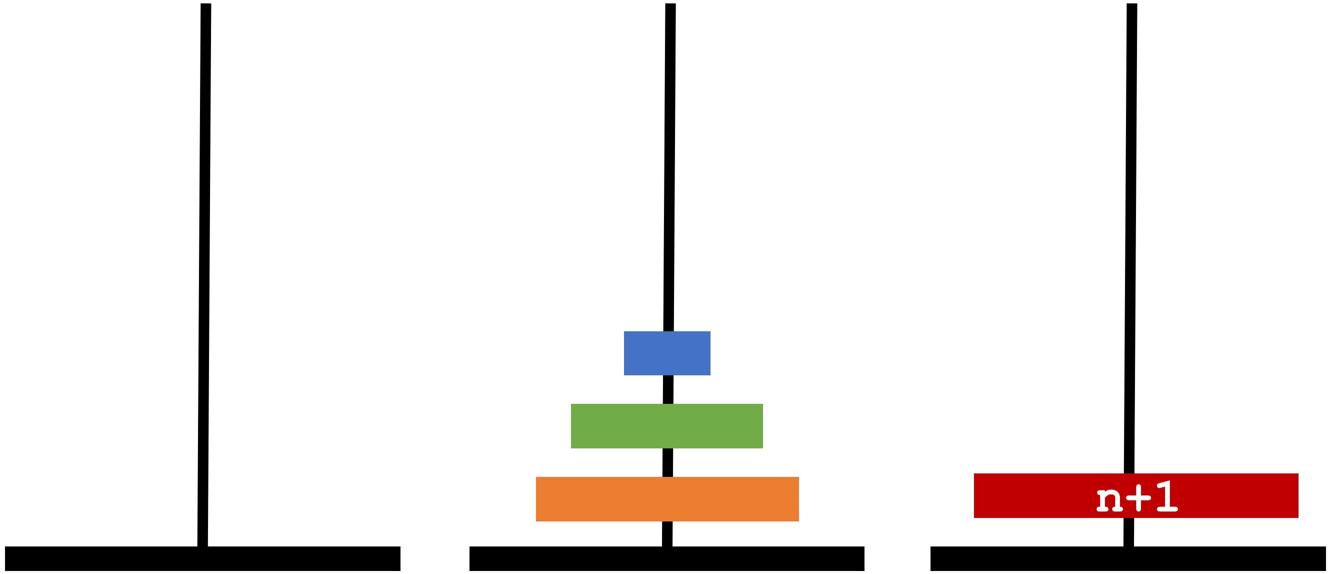
The Towers of Hanoi



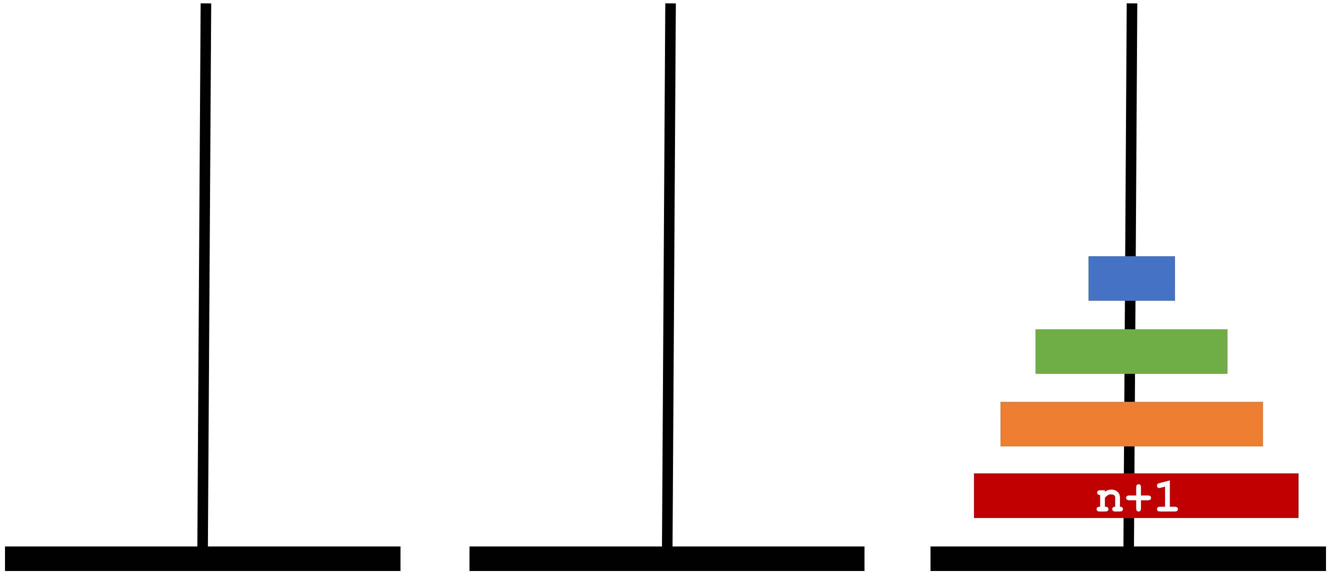
The Towers of Hanoi



The Towers of Hanoi



The Towers of Hanoi



The Towers of Hanoi

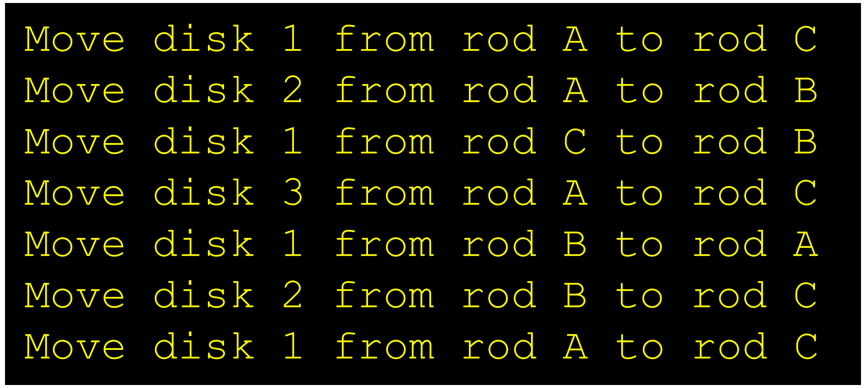
```
#include <iostream>
using namespace std;

void solveTowers(int n, char from_rod, char to_rod, char aux_rod)
{
          if (n == 1)
          {
              // Move from SRC to DST
              cout << "Move disk 1 from rod " << from_rod << " to rod " <<
              to_rod << endl;
              return;
          }

          solveTowers(n-1, from_rod, aux_rod, to_rod);
          // Move from SRC: FROM_ROD to DST: TO_ROD with SPARE: AUX_ROD
          cout << "Move disk " << n << " from rod " << from_rod << " to rod
" << to_rod << endl;

          // Move from SRC: AUX_ROD to DST: TO_ROD with SPARE: FROM_ROD
          solveTowers(n-1, aux_rod, to_rod, from_rod);
}

int main()
{
          int n = 3; // Number of disks
          solveTowers(n, 'A', 'C', 'B'); // A, B and C are names of rods
          return 0;
}
```

The Towers of Hanoi

```cpp
#include <iostream>
using namespace std;

void solveTowers(int n, char from_rod, char to_rod, char aux_rod)
{
          if (n == 1)
          {
              // Move from SRC to DST
              cout << "Move disk 1 from rod " << from_rod << " to rod " <<
              to_rod << endl;
              return;
          }

          solveTowers(n-1, from_rod, aux_rod, to_rod);
          // Move from SRC: FROM_ROD to DST: TO_ROD with SPARE: AUX_ROD
          cout << "Move disk " << n << " from rod " << from_rod << " to rod
" << to_rod << endl;

          // Move from SRC: AUX_ROD to DST: TO_ROD with SPARE: FROM_ROD
          solveTowers(n-1, aux_rod, to_rod, from_rod);
}

int main()
{
          int n = 3; // Number of disks
          solveTowers(n, 'A', 'C', 'B'); // A, B and C are names of rods
          return 0;
}
```

The Towers of Hanoi

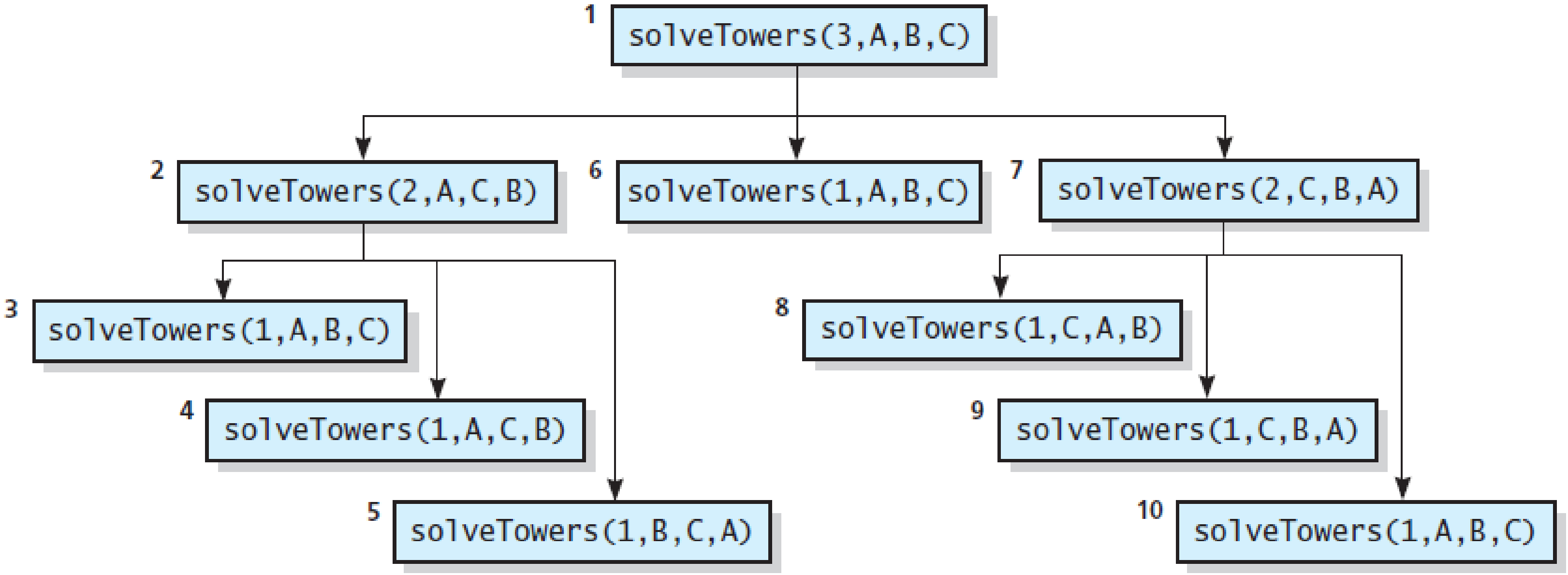
```
#include <iostream>
using namespace std;

void solveTowers(int n, char from_rod, char to_rod, char aux_rod)
{
          if (n == 1)
          {
              // Move from SRC to DST
              cout << "Move disk 1 from rod " << from_rod << " to rod " <<
              to_rod << endl;
              return;
          }

          solveTowers(n-1, from_rod, aux_rod, to_rod);
          // Move from SRC: FROM_ROD to DST: TO_ROD with SPARE: AUX_ROD
          cout << "Move disk " << n << " from rod " << from_rod << " to rod
" << to_rod << endl;

          // Move from SRC: AUX_ROD to DST: TO_ROD with SPARE: FROM_ROD
          solveTowers(n-1, aux_rod, to_rod, from_rod);
}

int main()
{
          int n = 3; // Number of disks
          solveTowers(n, 'A', 'C', 'B'); // A, B and C are names of rods
          return 0;
}
```



```
Move disk 1 from rod A to rod C
Move disk 2 from rod A to rod B
Move disk 1 from rod C to rod B
Move disk 3 from rod A to rod C
Move disk 1 from rod B to rod A
Move disk 2 from rod B to rod C
Move disk 1 from rod A to rod C
```

The Towers of Hanoi

- Order of recursive calls that results from `solveTowers(3,A,B,C)`



Thank you