

Towers of Hanoi

Move n disks from post 1 to post 2

- = Move n-1 disks from post 1 to post 3
- + Move 1 disk from post 1 to post 2
- + Move n-1 disks from post 3 to post 2

	1 2 3		
Move 3 disks from post 1 to post 2			
Move 2 disks from post 1 to post 3			
Move 1 disk from post 1 to post 2	2 3	1	
Move 1 disk from post 1 to post 3	3	1	2
Move 1 disk from post 2 to post 3	3		1 2
Move 1 disk from post 1 to post 2		3	1 2
Move 2 disks from post 3 to post 2			
Move 1 disk from post 3 to post 1	1	3	2
Move 1 disk from post 3 to post 2	1	2 3	
Move 1 disk from post 1 to post 2		1 2 3	

Move(n, i, j)

- = Let post k be the post other than i or j
- Move(n-1, i, k)
- Move(1, i, j)
- Move(n-1, k, j)

	Number of moves	
1 disk	$T[1] = 1$	1
2 disks	$T[2] = 2 * T[1] + 1$	3
3 disks	$T[3] = 2 * T[2] + 1$	7
4 disks	$T[4] = 2 * T[3] + 1$	15
n disks	$T[n] = 2 * T[n-1] + 1$	

Let $P(n)$ be the proposition $T[n] = 2^k - 1$

$P(1)$ is true

Assume $P(n)$ is true for $n = k$

$$\begin{aligned}
 T[k+1] &= 2 * T[k] + 1 \\
 &= 2 * (2^k - 1) + 1 \\
 &= 2^{k+1} - 1
 \end{aligned}$$

Then $P(n)$ is true for $n = k+1$

By induction, $P(n)$ is true for all $n \geq 1$

64 disks at 1 move/second = 585 billion years (age of the universe = 14 billion years)

where 1 billion = 1,000 million

```

// Towers of Hanoi

#include <iostream>

using namespace std;

#define IN
#define OUT

void Move( IN int n, IN int pegA, IN int pegB );

void main() {
    int n = 4;          // Number of disks
    Move( n, 1, 2 );   // Move n disks from peg 1 to peg 2
    system( "pause" );
}

// Move n disks from pegA to pegB

void Move( IN int n, IN int pegA, IN int pegB ) {
    if ( n == 1 )

```

```

    cout << "A disk is moved from peg " << pegA << " to " << pegB <<
endl;
    else {
        int pegC; // the peg that is neither pegA nor pegB

        // determine pegC

        switch ( pegA ) {
            case 1: pegC = (( pegB == 2 ) ? 3 : 2); break;
            case 2: pegC = (( pegB == 3 ) ? 1 : 3); break;
            case 3: pegC = (( pegB == 1 ) ? 2 : 1); break;
        }

        Move( n-1, pegA, pegC );
        Move( 1, pegA, pegB );
        Move( n-1, pegC, pegB );
    }
}

```

n = 3

A disk is moved from peg 1 to 2
A disk is moved from peg 1 to 3
A disk is moved from peg 2 to 3
A disk is moved from peg 1 to 2
A disk is moved from peg 3 to 1
A disk is moved from peg 3 to 2
A disk is moved from peg 1 to 2

n = 4

A disk is moved from peg 1 to 3
A disk is moved from peg 1 to 2
A disk is moved from peg 3 to 2
A disk is moved from peg 1 to 3
A disk is moved from peg 2 to 1
A disk is moved from peg 2 to 3
A disk is moved from peg 1 to 3
A disk is moved from peg 1 to 2
A disk is moved from peg 3 to 2
A disk is moved from peg 3 to 1
A disk is moved from peg 2 to 1
A disk is moved from peg 3 to 2
A disk is moved from peg 1 to 3
A disk is moved from peg 1 to 2
A disk is moved from peg 3 to 2