Tower of Hanoi: analysis of the call stack

https://en.wikipedia.org/wiki/Tower_of_Hanoi

Liceo G.B. Brocchi – Bassano del Grappa (VI)
Liceo Scientifico – opzione scienze applicate
Giovanni Mazzocchin

Recursive solution - pseudocode

```
hanoi(nDisks, initPeg, tempPeg, finalPeg):
    if nDisks == 0:
        return
    hanoi(n - 1, initPeg, finalPeg, tempPeg)
    print ‘move disk from’, initPeg, ‘to’, finalPeg
    hanoi(n - 1, tempPeg, initPeg, finalPeg)
```

What happens on the call stack?

```
n:3     init:A     temp:B     final:C
            external call
```

What happens on the call stack?

```
n:2    init:A    temp:C    final:B
        1st recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
n:1    init:A    temp:B    final:C
         1st recursive call
```

```
n:2    init:A    temp:C    final:B
         1st recursive call
```

```
n:3    init:A    temp:B    final:C
             external call
```

What happens on the call stack?

```
n:0    init:A    temp:C    final:B
  1st recursive call (base case)
```

```
n:1    init:A    temp:B    final:C
        1st recursive call
```

```
n:2    init:A    temp:C    final:B
        1st recursive call
```

```
n:3    init:A    temp:B    final:C
            external call
```

What happens on the call stack?

```
print ‘move disk from A to C’
```

```
n:1    init:A    temp:B    final:C
 1st recursive call, make 2nd call
```

```
n:2    init:A    temp:C    final:B
        1st recursive call
```

```
n:3    init:A    temp:B    final:C
            external call
```

What happens on the call stack?

```
n:0    init:B    temp:A    final:C
  2nd recursive call (base case)
```

```
n:1    init:A    temp:B    final:C
        1st recursive call
```

```
n:2    init:A    temp:C    final:B
        1st recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
n:1    init:A    temp:B    final:C
         1st recursive call
```

```
n:2    init:A    temp:C    final:B
         1st recursive call
```

```
n:3    init:A    temp:B    final:C
             external call
```

What happens on the call stack?

```
print ‘move disk from A to B’
```

n:2 init:A temp:C final:B
1st recursive call, make 2nd call

n:3 init:A temp:B final:C
external call

What happens on the call stack?

```
n:1     init:C     temp:A     final:B
          2nd recursive call
```

```
n:2     init:A     temp:C     final:B
          1st recursive call
```

```
n:3     init:A     temp:B     final:C
              external call
```

What happens on the call stack?

```
n:0    init:C    temp:B    final:A
  1st recursive call (base case)
```

```
n:1    init:C    temp:A    final:B
        2nd recursive call
```

```
n:2    init:A    temp:C    final:B
        1st recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
print ‘move disk from C to B’
```

n:1 init:C temp:A final:B
2nd recursive call, make 2nd call

n:2 init:A temp:C final:B
1st recursive call

n:3 init:A temp:B final:C
external call

What happens on the call stack?

```
n:0    init:A    temp:C    final:B
  2nd recursive call (base case)
```

```
n:1    init:C    temp:A    final:B
        2nd recursive call
```

```
n:2    init:A    temp:C    final:B
        1st recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
n:1     init:C     temp:A     final:B
          2nd recursive call
```

```
n:2     init:A     temp:C     final:B
          1st recursive call
```

```
n:3     init:A     temp:B     final:C
              external call
```

What happens on the call stack?

```
n:2     init:A     temp:C     final:B
          1st recursive call
```

```
n:3     init:A     temp:B     final:C
              external call
```

What happens on the call stack?

```
print ‘move disk from A to C’
```

n:3 init:A temp:B final:C
external call, make 2nd call

What happens on the call stack?

```
n:2    init:B    temp:A    final:C
        2nd recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
n:1     init:B     temp:C     final:A
          1st recursive call
```

```
n:2     init:B     temp:A     final:C
          2nd recursive call
```

```
n:3     init:A     temp:B     final:C
              external call
```

What happens on the call stack?

```
n:0    init:B    temp:A    final:C
  1st recursive call (base case)
```

```
n:1    init:B    temp:C    final:A
        1st recursive call
```

```
n:2    init:B    temp:A    final:C
        2nd recursive call
```

```
n:3    init:A    temp:B    final:C
            external call
```

What happens on the call stack?

```
print ‘move disk from B to A’
```

```
n:1    init:B    temp:C    final:A
 1st recursive call, make 2nd call
```

```
n:2    init:B    temp:A    final:C
        2nd recursive call
```

```
n:3    init:A    temp:B    final:C
            external call
```

What happens on the call stack?

```
n:0    init:C    temp:B    final:A
  2nd recursive call (base case)
```

```
n:1    init:B    temp:C    final:A
 1st recursive call, make 2nd call
```

```
n:2    init:B    temp:A    final:C
         2nd recursive call
```

```
n:3    init:A    temp:B    final:C
             external call
```

What happens on the call stack?

```
n:1     init:B     temp:C     final:A
          1st recursive call
```

```
n:2     init:B     temp:A     final:C
          2nd recursive call
```

```
n:3     init:A     temp:B     final:C
              external call
```

What happens on the call stack?

```
print ‘move disk from B to C’
```

n:2 init:B temp:A final:C
2nd recursive call, make 2nd call

n:3 init:A temp:B final:C
external call

What happens on the call stack?

```
n:1     init:A     temp:B     final:C
          2nd recursive call
```

```
n:2     init:B     temp:A     final:C
          2nd recursive call
```

```
n:3     init:A     temp:B     final:C
              external call
```

What happens on the call stack?

```
n:0    init:B    temp:A    final:C
  1st recursive call (base case)
```

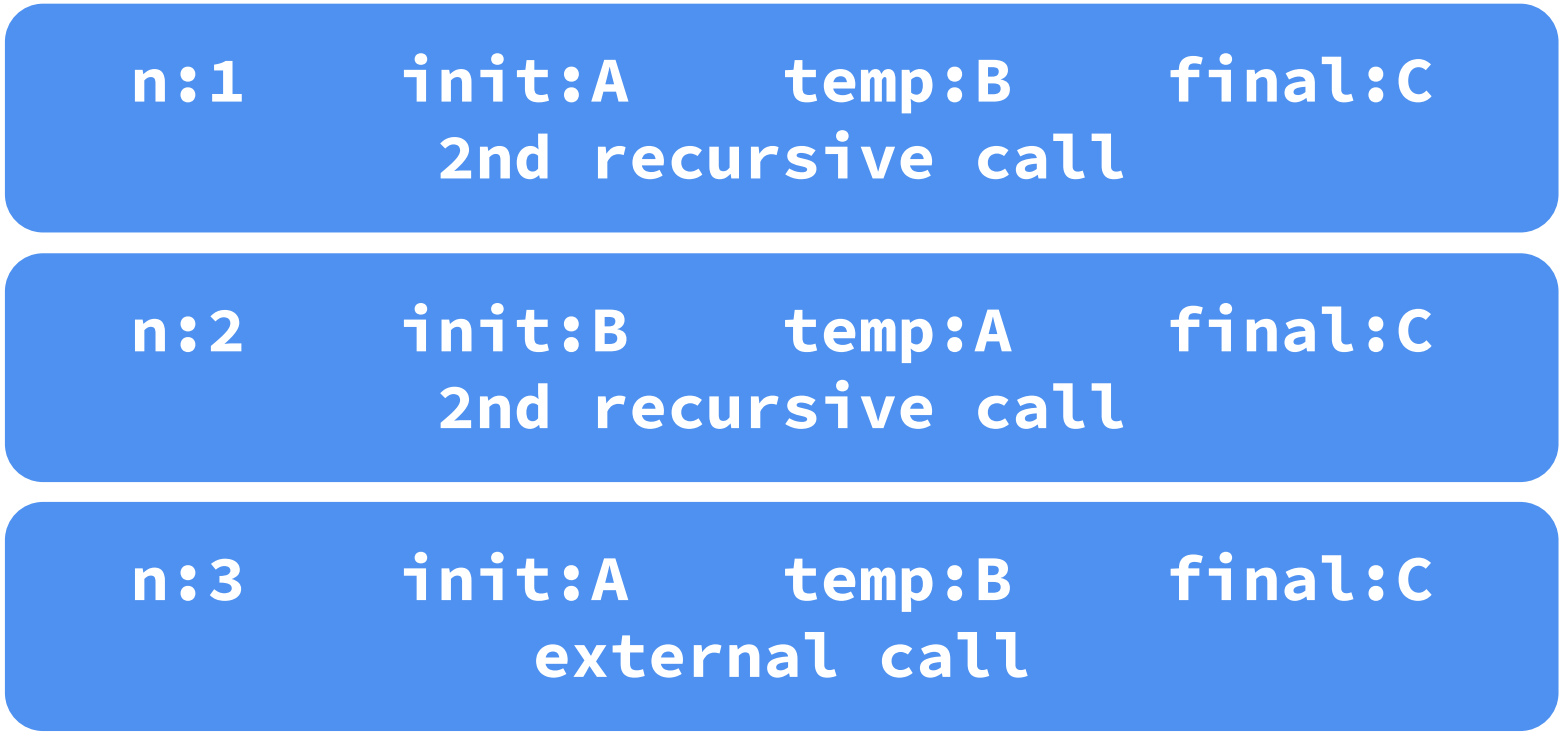
```
n:1    init:A    temp:B    final:C
        2nd recursive call
```

```
n:2    init:B    temp:A    final:C
        2nd recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
print ‘move disk from A to C’
```



What happens on the call stack?

```
n:2    init:B    temp:A    final:C
        2nd recursive call
```

```
n:3    init:A    temp:B    final:C
           external call
```

What happens on the call stack?

```
n:3    init:A    temp:B    final:C
  external call, return to caller
```

Links

- Recursion 'Super Power' (in Python) - Computerphile