Tower of Hanoi: a low-level analysis of the recursive solution's call stack

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

Liceo G.B. Brocchi
Classi seconde Scientifico – opzione scienze applicate
Bassano del Grappa, Maggio 2023
Prof. Giovanni Mazzocchin

Recursive solution - pseudocode

```
hanoi(nDisks, initPeg, tempPeg, finalPeg):
    if nDisks == 1:
        print ‘move disk from’, initPeg,  ‘to’, finalPeg
        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 (3 disks)

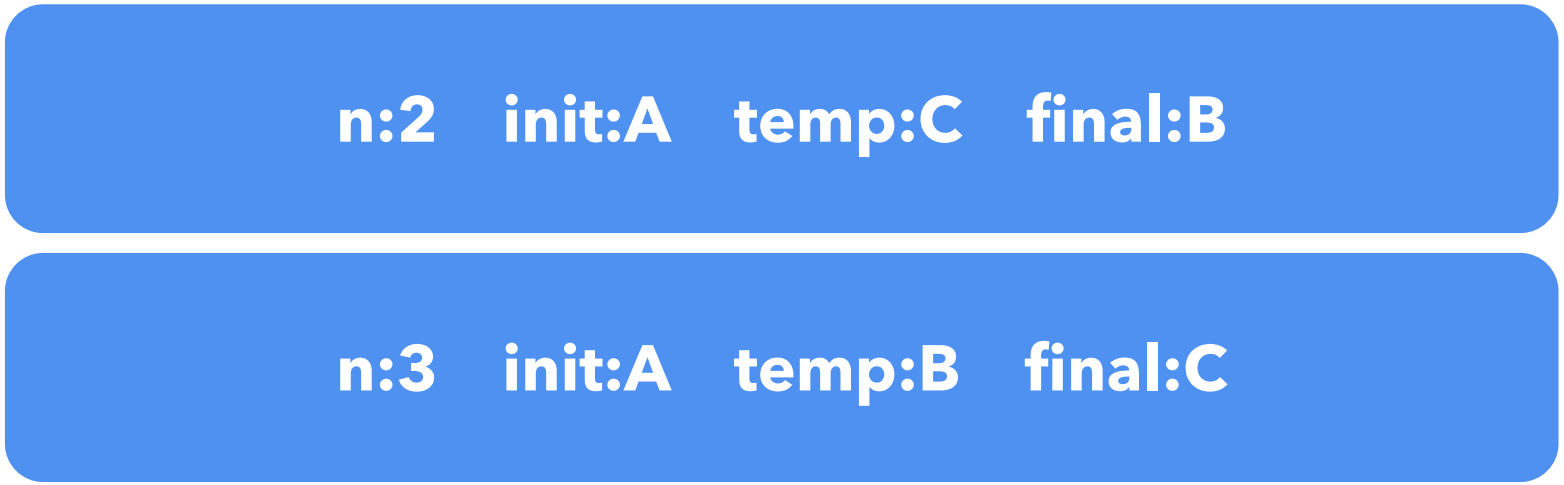
CALL STACK

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

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK

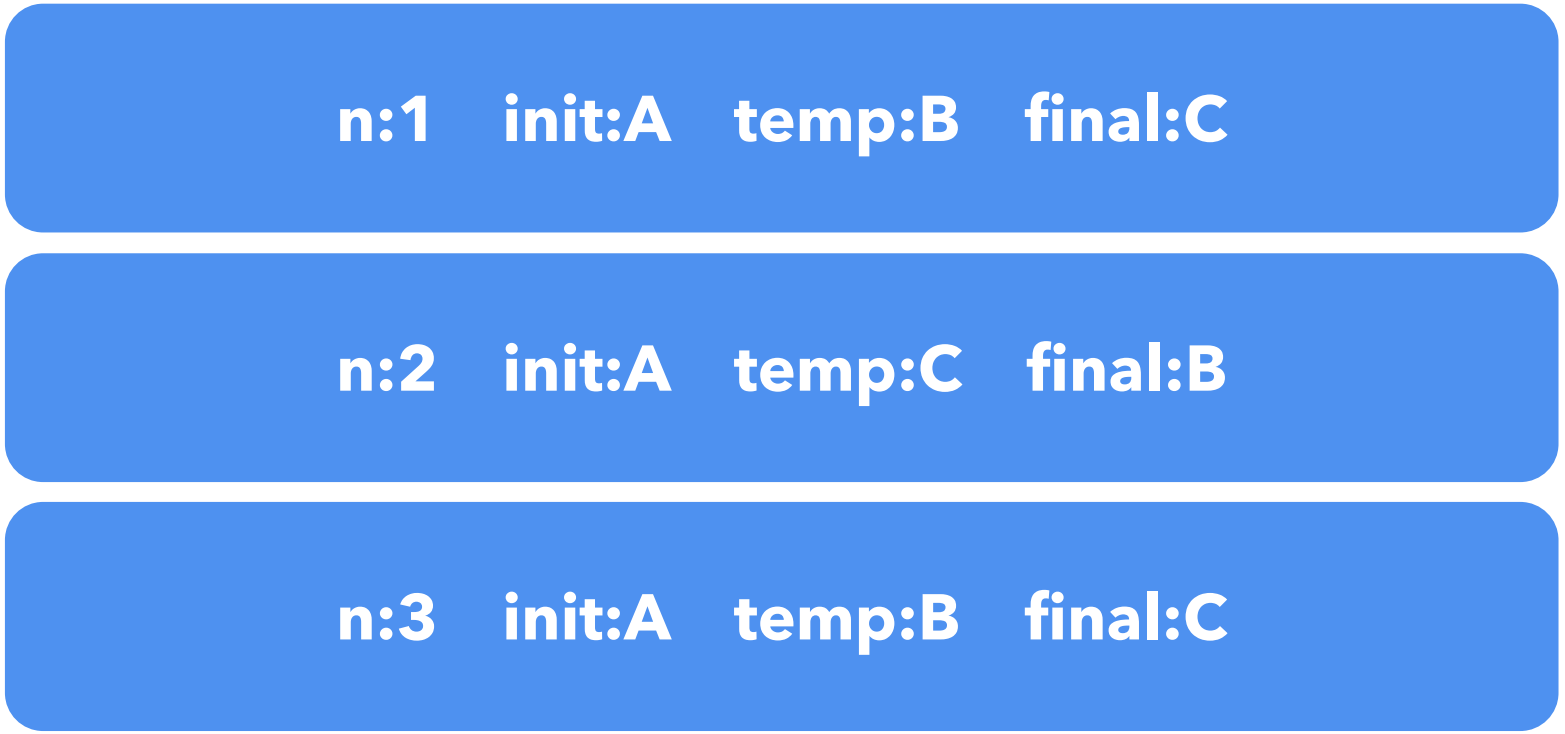


1st call's stack frame;

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK



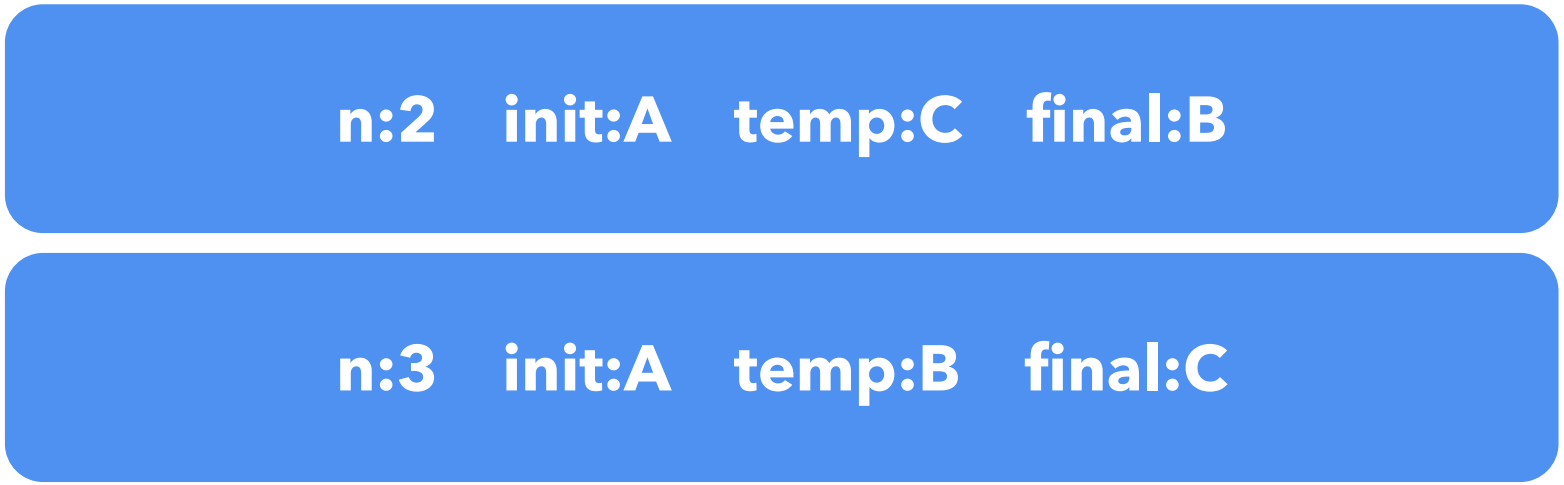
1st call's stack frame;
print 'A ->C'; base case: return and pop the stack frame;

1st call's stack frame;

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK

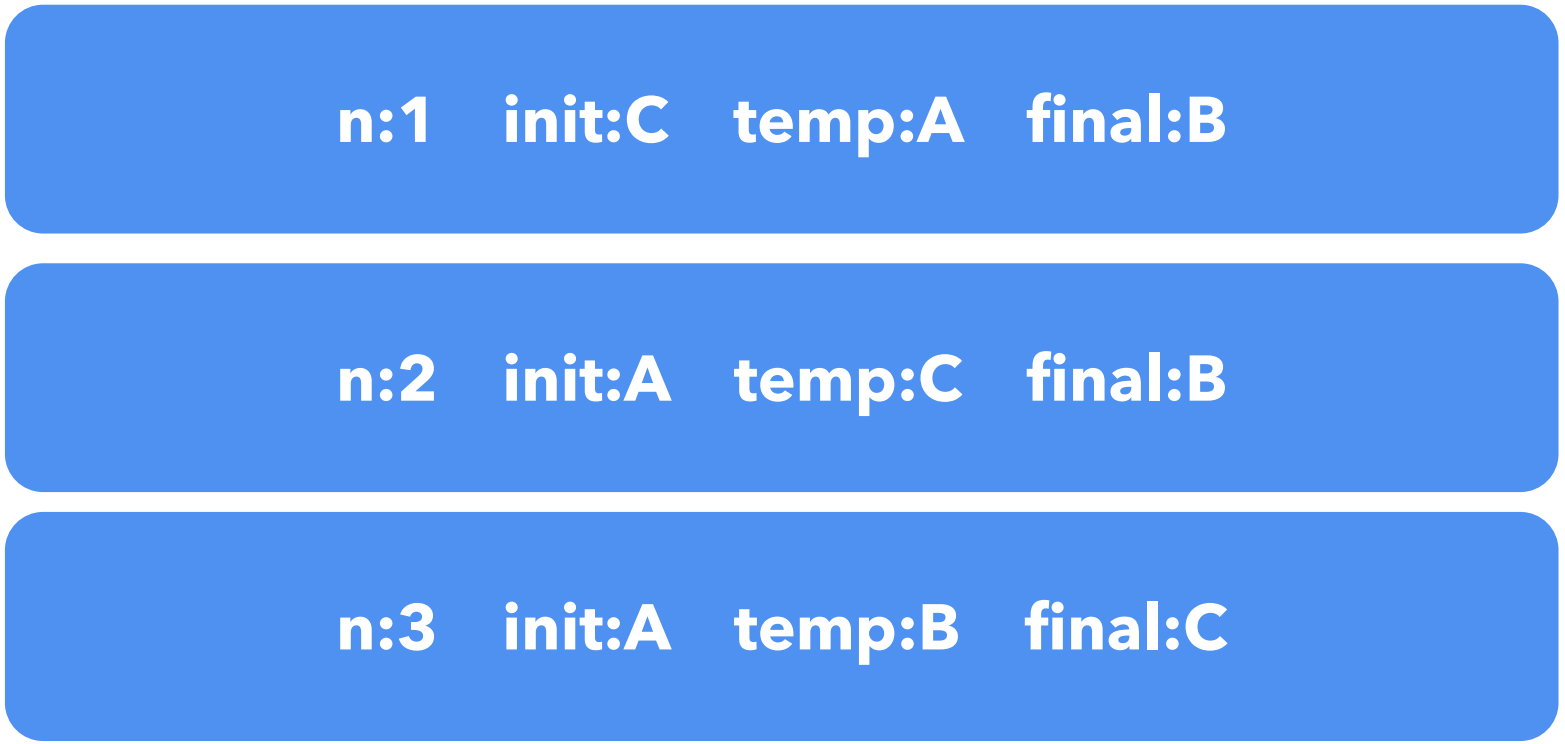


1st call's stack frame;
just returned from 1st call;
print 'A->B'; make 2nd call;

external call's stack frame

What happens on the call stack (3 disks)

CALL STACK



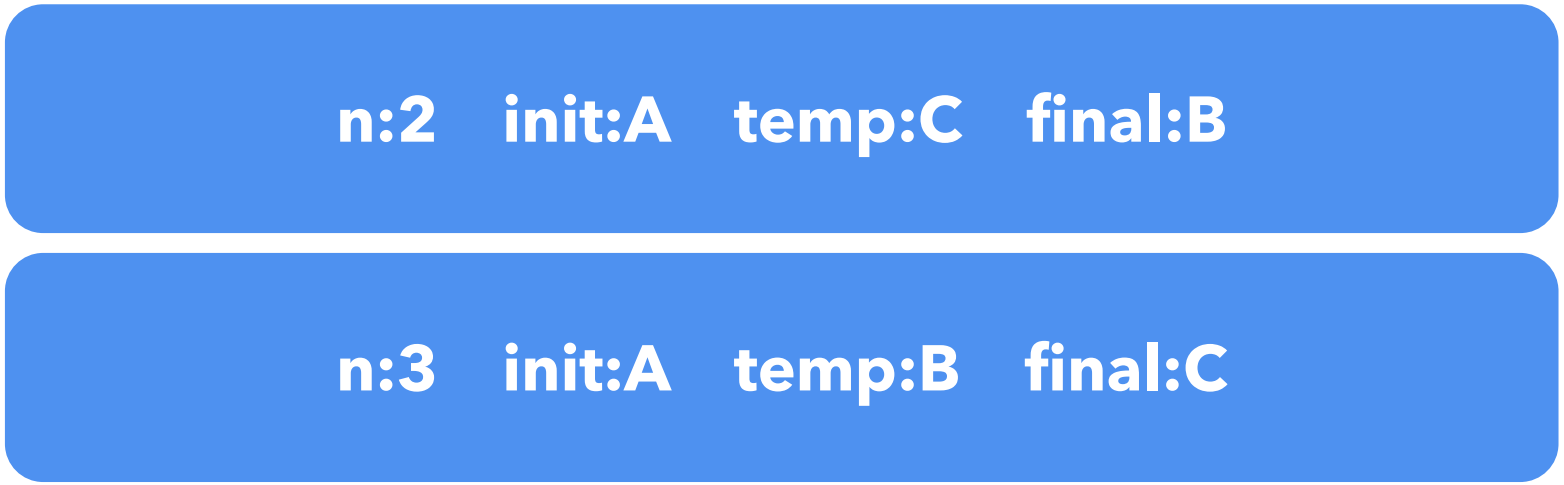
2nd call's stack frame; print 'C->B'; base case: return and pop the stack frame;

1st call's stack frame;

external call's stack frame

What happens on the call stack (3 disks)

CALL STACK



1st call's stack frame;
just returned from 2nd call;
nothing to do, just return and pop the stack frame;

external call's stack frame

What happens on the call stack (3 disks)

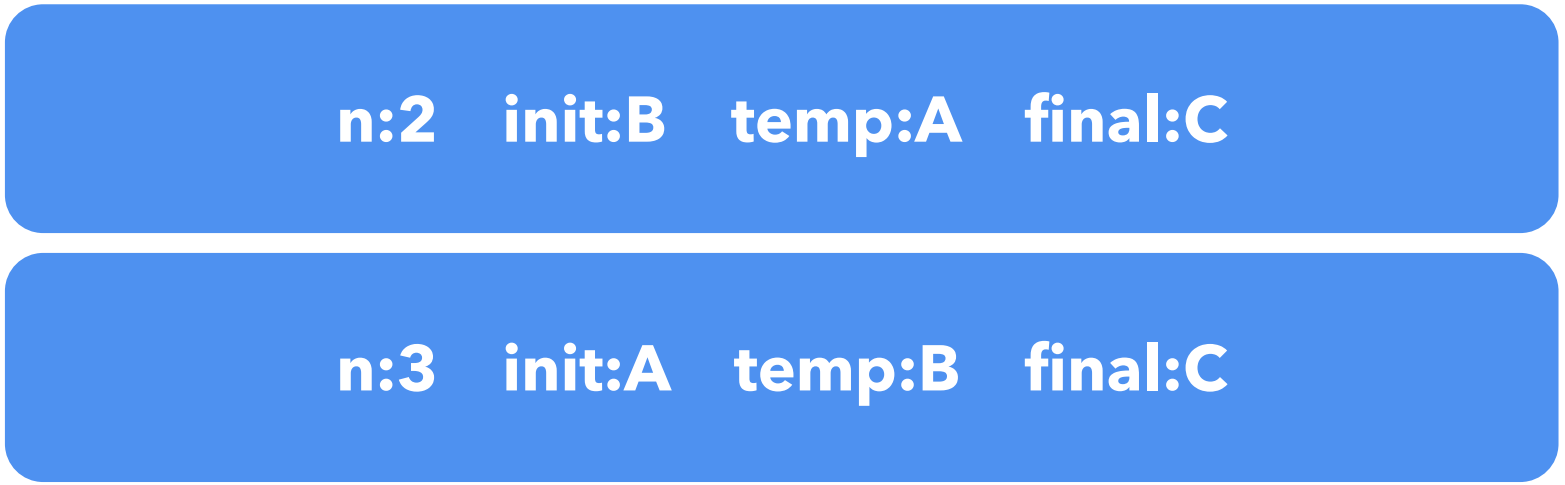
CALL STACK

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

external call's stack frame;
just returned from 1st call;
print 'A->C'; make 2nd call;

What happens on the call stack (3 disks)

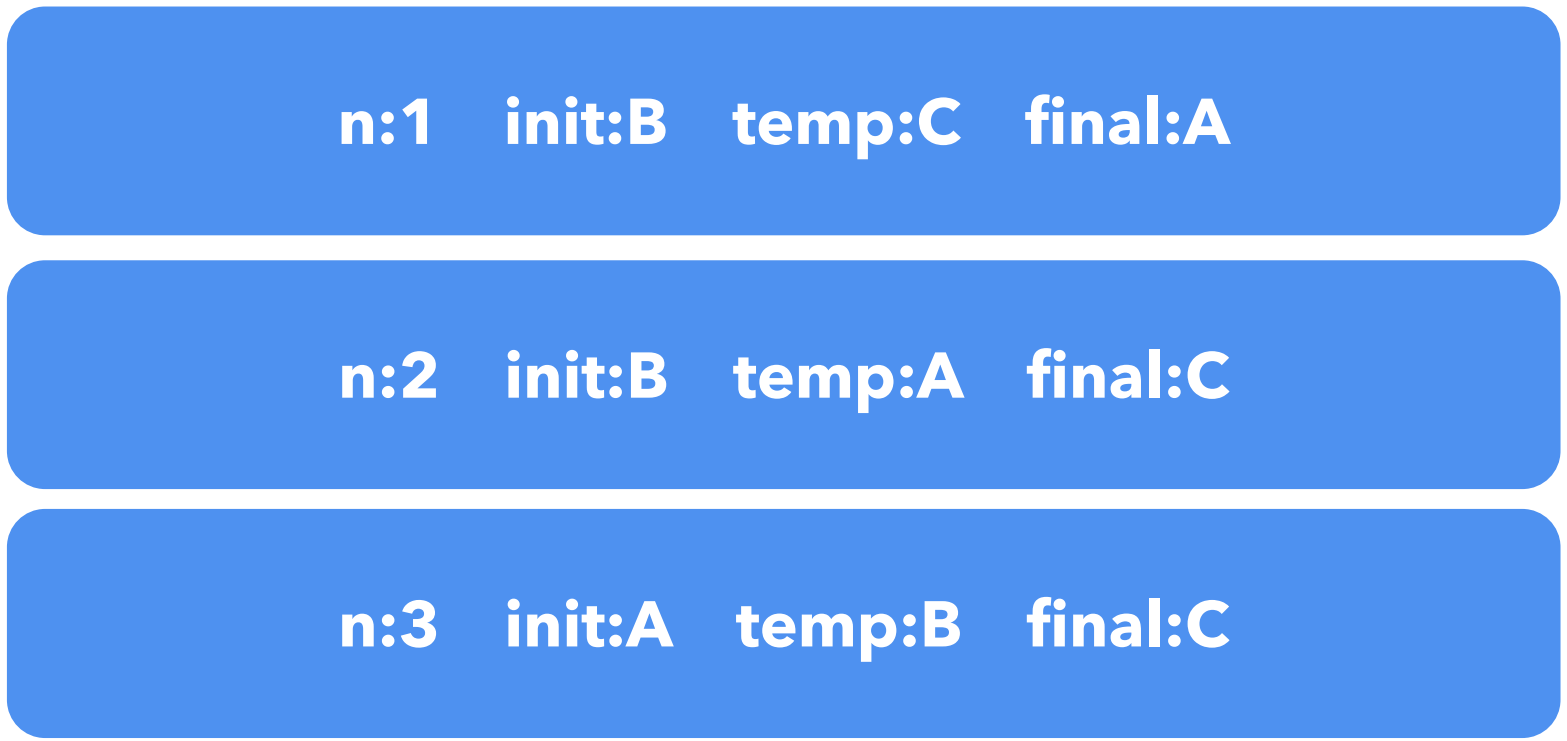
CALL STACK



2nd call's stack frame;

external call's stack frame;

What happens on the call stack (3 disks)



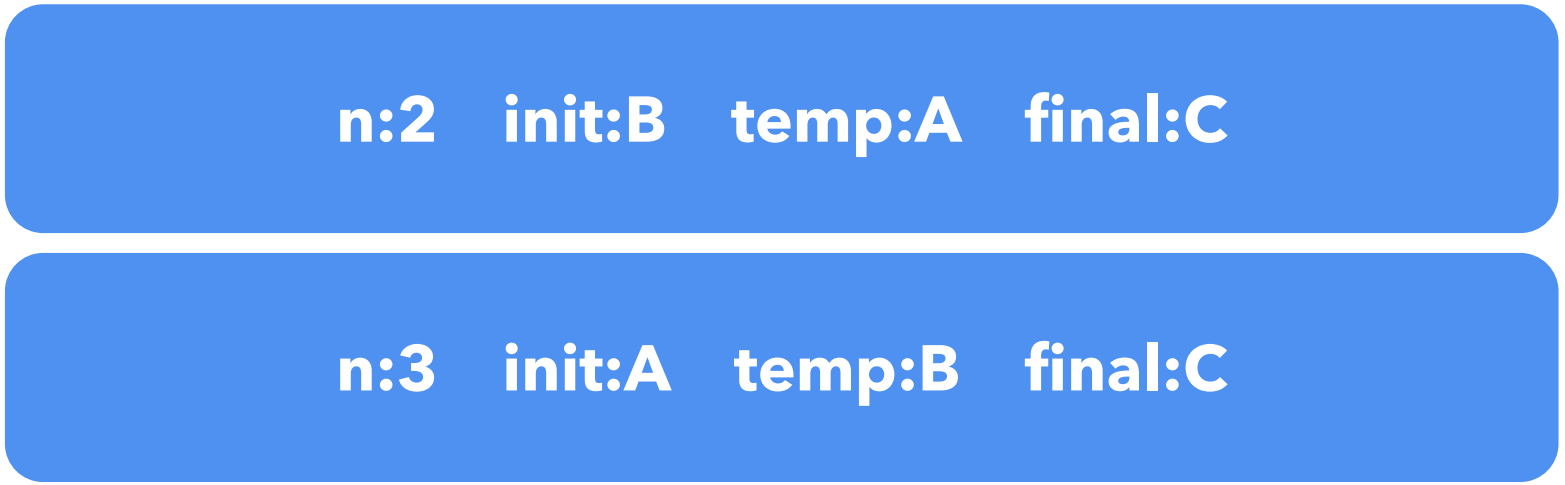
1st call's stack frame;
print 'B->A'; base case: return and pop the stack frame;

2nd call's stack frame;

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK

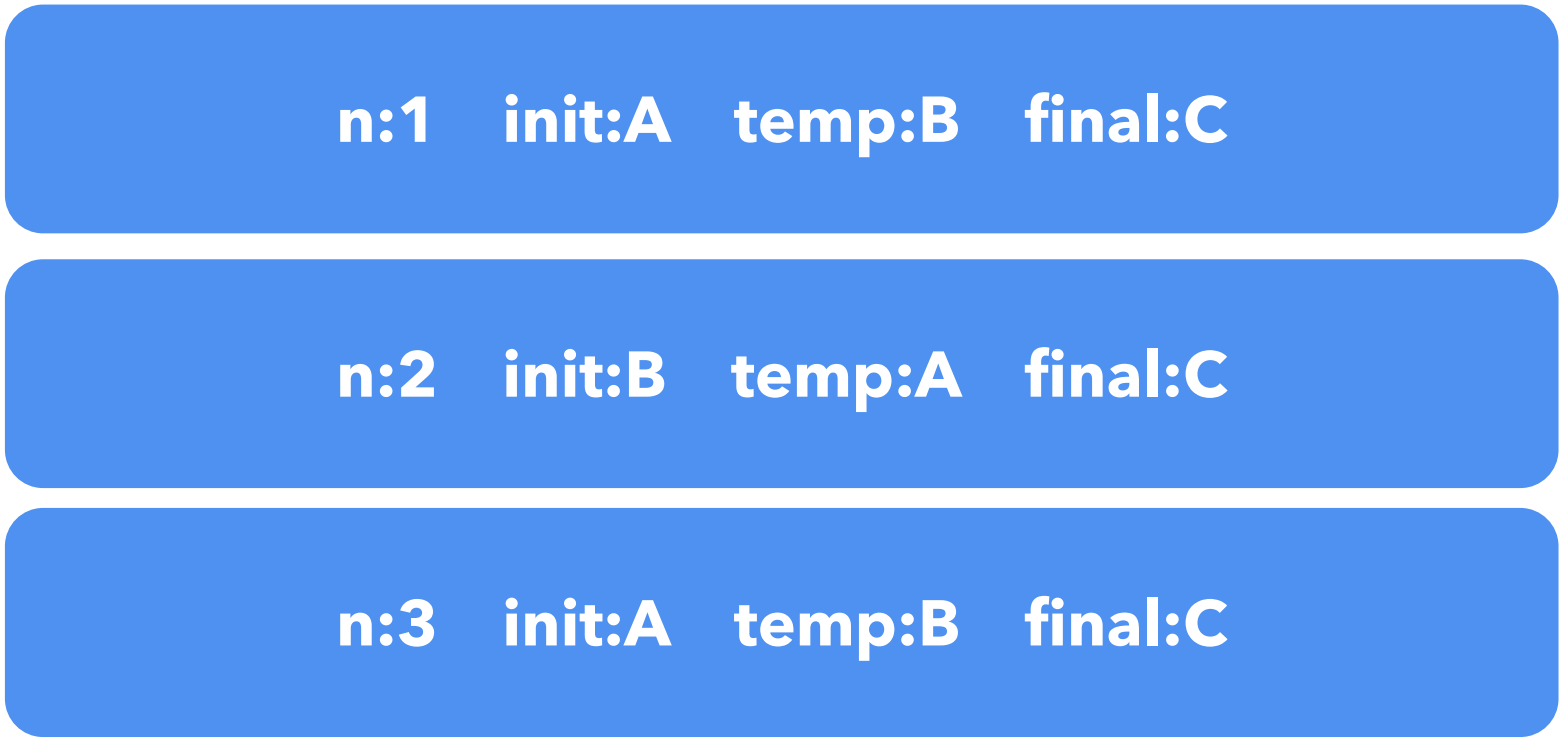


2nd call's stack frame; just returned from 1st call; print 'B->C'; make 2nd call;

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK



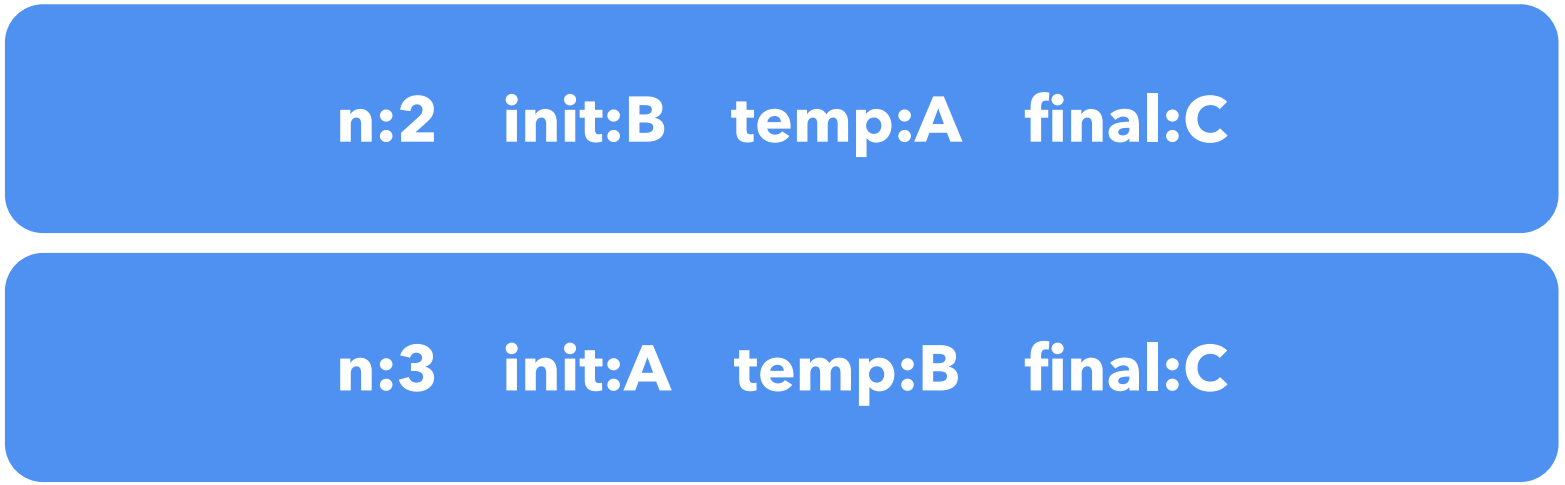
2nd call's stack frame; print 'A->C'; base case: return and pop the stack frame;

2nd call's stack frame;

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK



2nd call's stack frame;

external call's stack frame;

What happens on the call stack (3 disks)

CALL STACK

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

external call's stack frame;
just returned from 2nd call;
pop the stack frame and return
to the external caller;