

3 Sept 2021

Proofs of Correctness

crux: defining your invariants.

→ something that is "always" true.

→ for POC, we can think of these as assert statements.

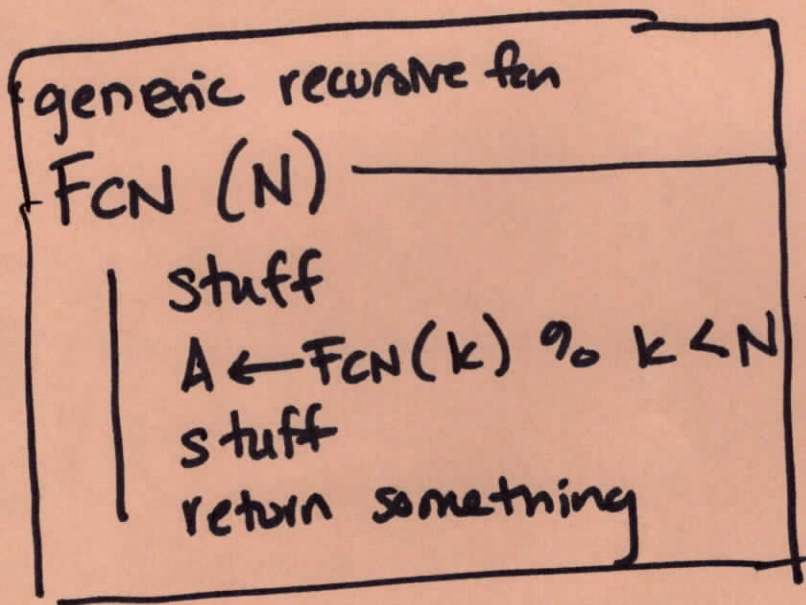
→ statement, must evaluate to TRUE or FALSE.

Initialization

Maintenance

End (Termination)

* Be cautious - this is not showing that it terminates! d



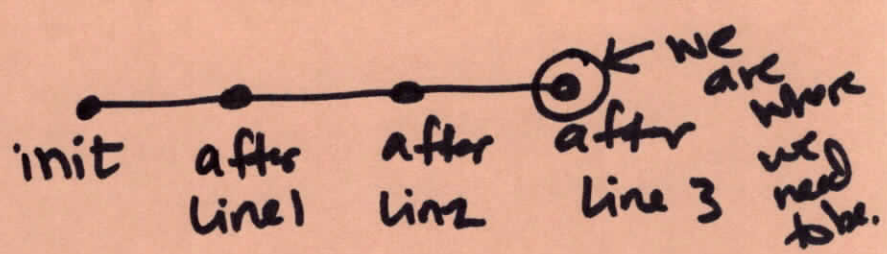
END: ^{our top level} If FCN(N) returns,
then it returns the right thing.

Maintenance: If calls to smaller cases return
the right thing, then I return
the right thing.

generic serial fn

FCNB(A, B, C)

- 1: stuff
- 2: more stuff
- 3: returns something



e.g.,

average(b, a)

$$b' \leftarrow \frac{b}{2}$$

$$a' \leftarrow \frac{a}{2}$$

$$\text{return } a' + b'$$

Proof that average(b, a) is correct

After line 1, we have

$$\left. \begin{array}{l} a = a \\ b = b \\ b' = \frac{b}{2} \end{array} \right\} \text{inputs}$$

After line 2, we also have $a' = \frac{a}{2}$

what is returned is

$$a' + b' = \frac{a}{2} + \frac{b}{2} = \frac{a+b}{2},$$

as was to be shown \square

The tricky part comes when we encounter loops and recursion. Then, our proofs of correctness start to look a lot like induction!

INITIALIZATION: $FCN(1)$ returns the right value.
↑ or whatever our base cases are
+ recursion invariant is ~~true~~ true upon return.
Often very simple to prove.
Like our base case in induction

~~THE~~

HANOI (N , src, dst, tmp)

```
1: if  $|N| \leq 1$ 
2: | move disk from src to dst.
3: else
4: | HANOI ( $N-1$ , src, tmp, dst)
5: | move last disk from src to dst
6: | HANOI ( $N-1$ , tmp, dst, src)
7: end if
8: return
```

ASSERT

My recursion invariant: an assert at the end of a recursion

R1: there are currently no "violations" of smaller disks on larger disks.

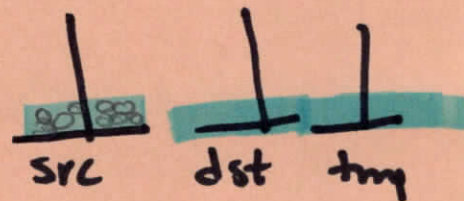
R2: the top N disks that were on src are now on dst.

at line 1.

③

consider $n=1$.

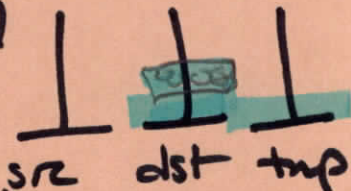
Hanoi(1, src, dst, tmp)



Then, in line 2, the disk is moved

from the src to the dst.

I can do this b/c it is the smallest.



Lines 3-7 are in the else, so do not run

So, at line 8, when we return that recursion invariant is true!

Initial assumptions:

→ when I call the 1st case,

all N disks are on src. (dst & tmp

do not have any disks.

Let $k \geq 1$.

Maintenance: If the recursion inv. holds for k , then it holds for $k+1$.

↖ note: this is just like induction!

Proof: ~~by induction~~

Note $k+1 > 1$, therefore we enter the else in Line 3.

After line 4, by R.I., I have

① No violations

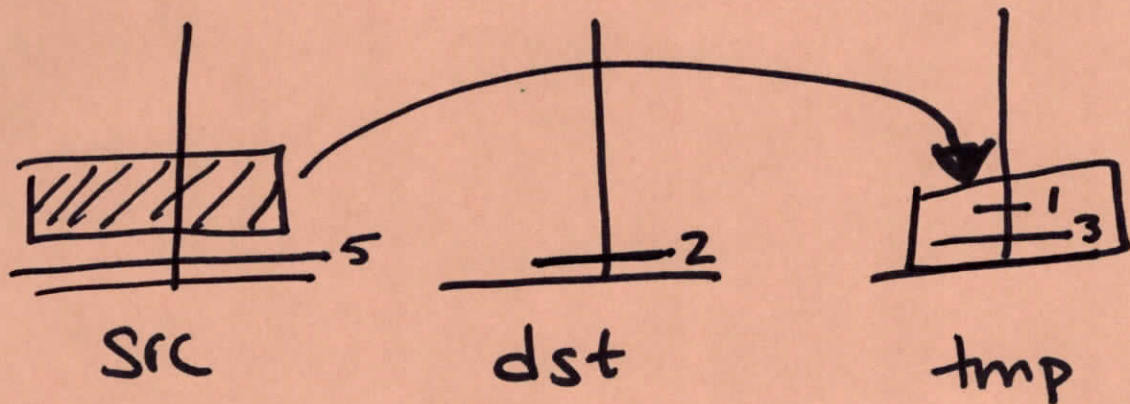
② the smallest ~~now~~ $k+1-1=k$ disks that were on S_2 are now on ~~src~~ ^{tmp}.

<thinking>

Analogy to
Saying "by the
inductive assumption"

<thinking>

Why is this not the case in line 5?



Ans: We don't have everything we need in that recursion inv.
So add another clause to it!

R3 ○