For those that haven't seen it, an example proof by induction:

Claim: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \quad \forall n \in \mathbb{Z}^+$.

Proof (by induction):

① Establish a "base case" explicitly.

if $n=1$, then $\sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2}$ ✓

② Assume truth for $n-1$. Show this implies truth for n.

Break down $\sum_{i=1}^{n} i$ into smaller pieces so we can use the assumption

$$\sum_{i=1}^{n} i = \boxed{\sum_{i=1}^{n-1} i} + n$$

$$= \frac{(n-1)n}{2} + n = \frac{(n-1)n + 2n}{2}$$

$$= \frac{n(n-1+2)}{2}$$

$$= \frac{n(n+1)}{2}$$ Yay!!

① $T(1)$ holds.

② $T(n-1) \Rightarrow T(n)$.

This suffices to show $T(n) \; \forall \; n \geq 1$.

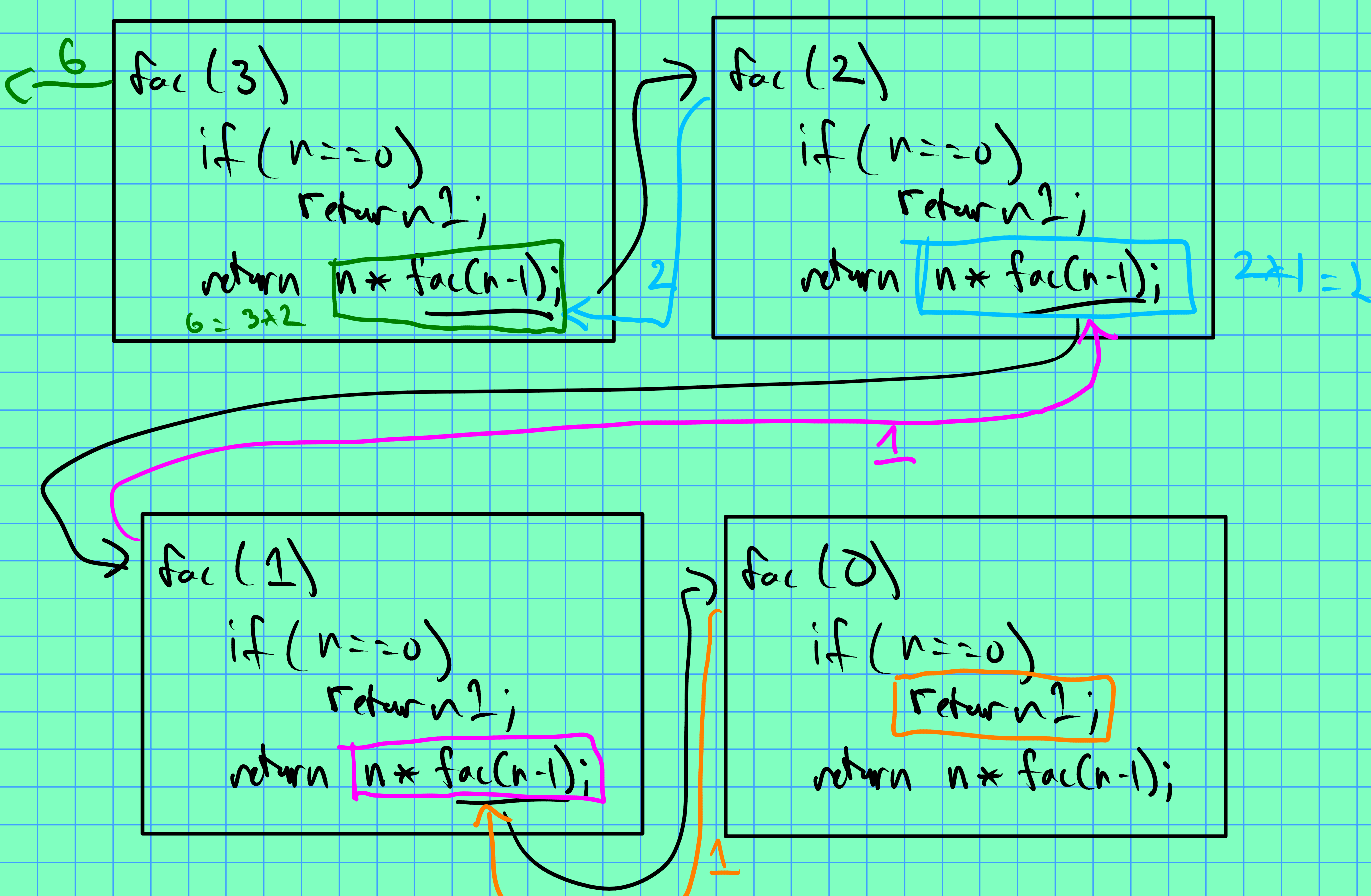
E.g., if $n = 5$:

$$\underset{①}{T(1)} \xRightarrow{②} T(2) \xRightarrow{②} T(3) \xRightarrow{②} T(4) \xRightarrow{②} T(5) \checkmark$$

Similar example in code: write a recursive function to compute $n!$

```
int fac(int n) {
    if (n==0)
        return 1;   // ①
    // Now assume T(n-1), i.e., our fac(.)
    // works on n-1. So, fac(n-1) = (n-1)!
    // note:  n! = n (n-1)!
    return n * fac(n-1);
}
```

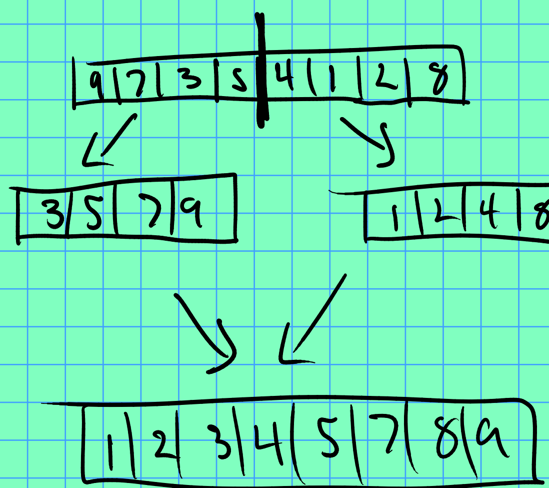
Let's trace this on $n = 3$:



Example 2: Recursive sorting.

Idea: (a) break array into 2 pieces that are <u>smaller</u>, & sort them recursively.

(b) merge two sorted arrays together



(a)

(b)