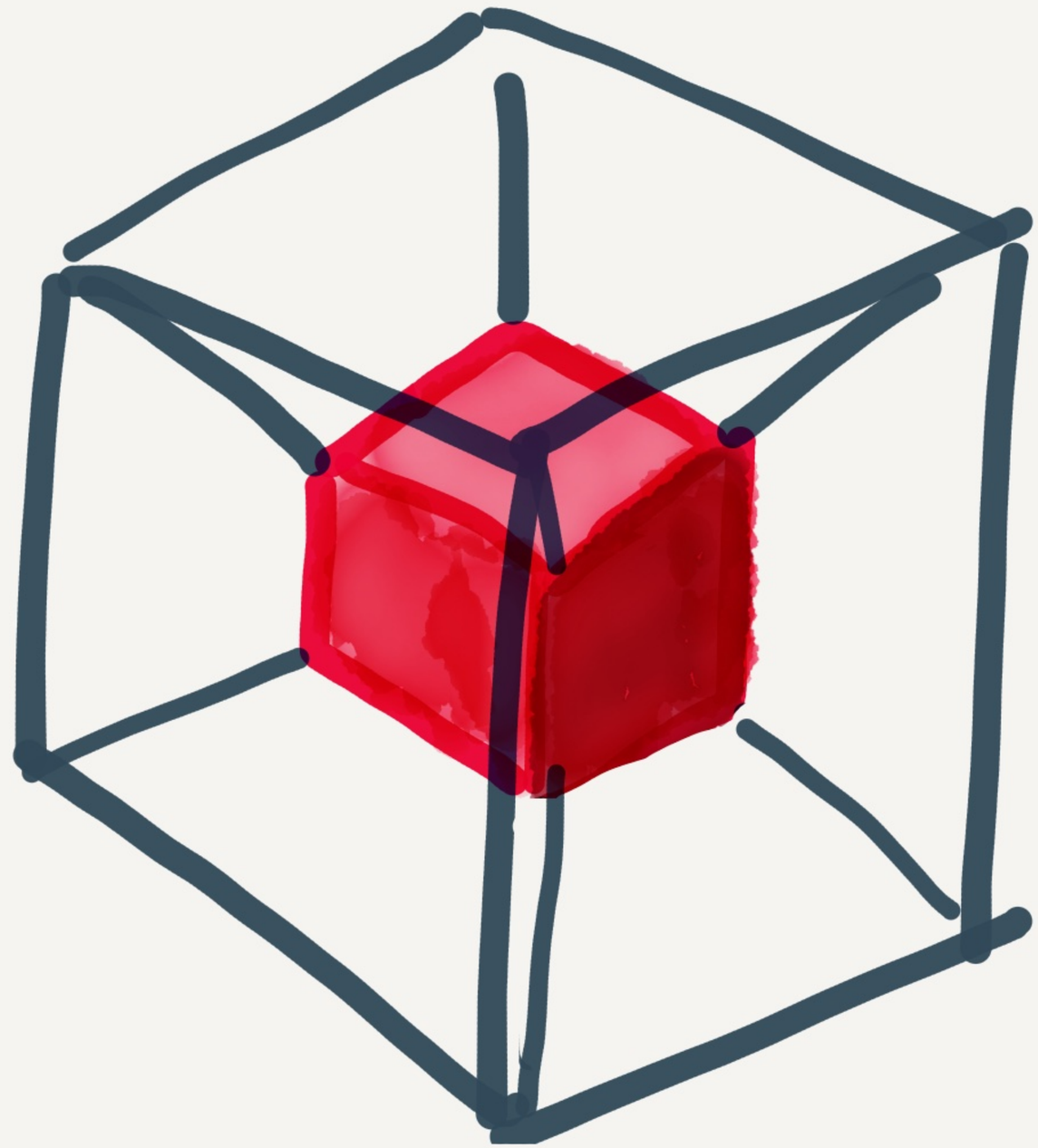
in

TIME

a

WRINKLE



tesser

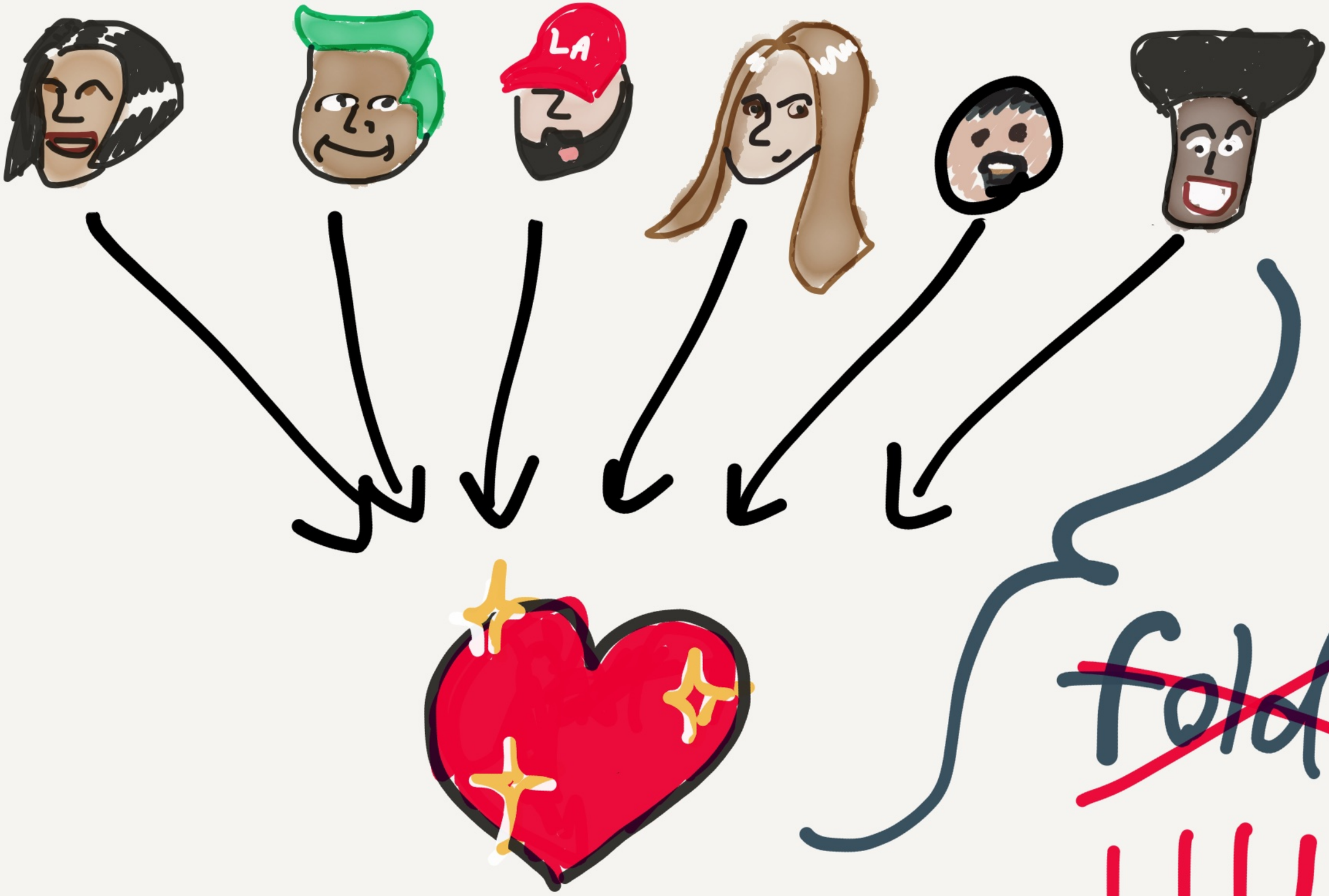


"aphyr"

Kyle Kingsbury

[jepsen.io]

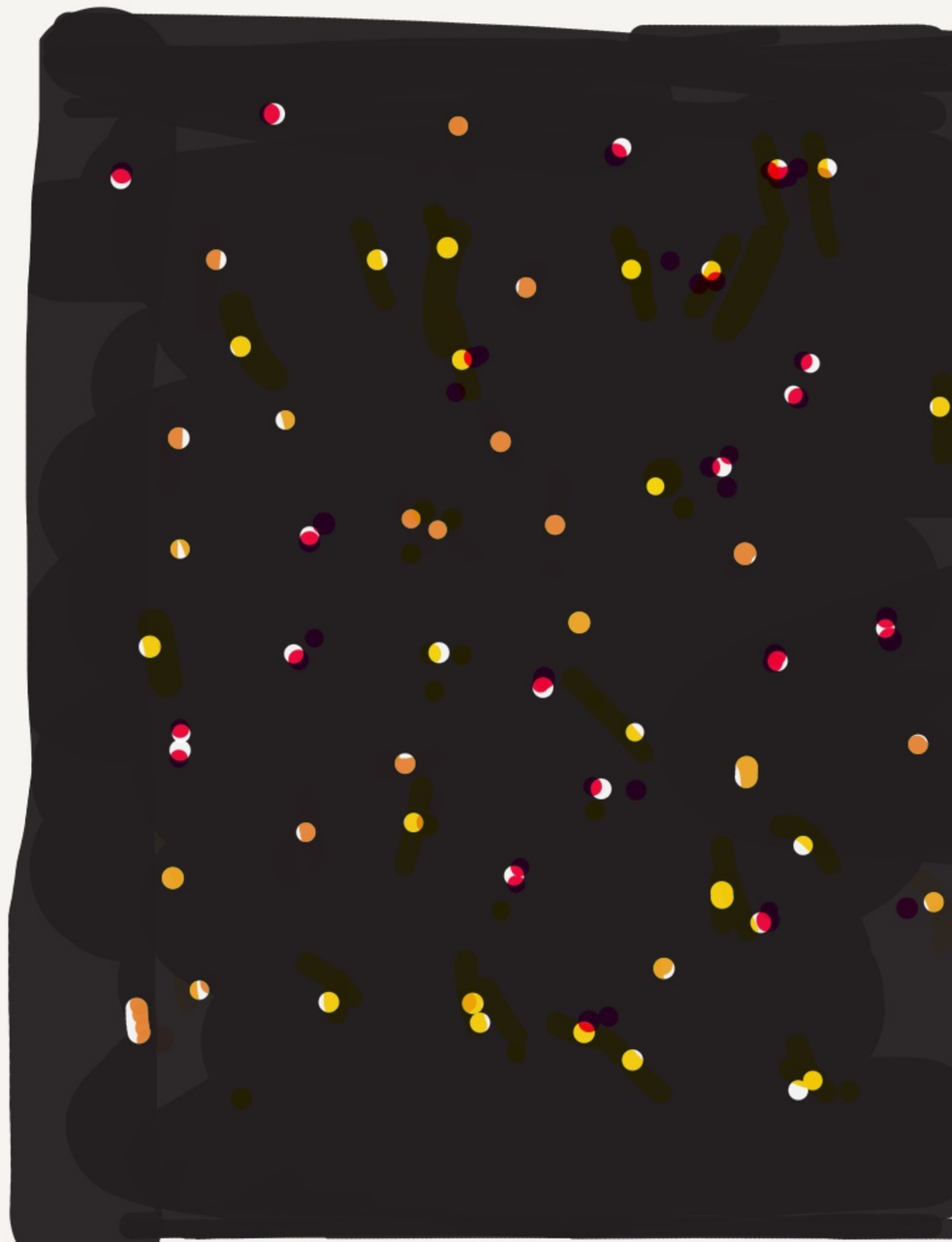
What is it LIKE to be a FOLD?


LA
fold
HUG

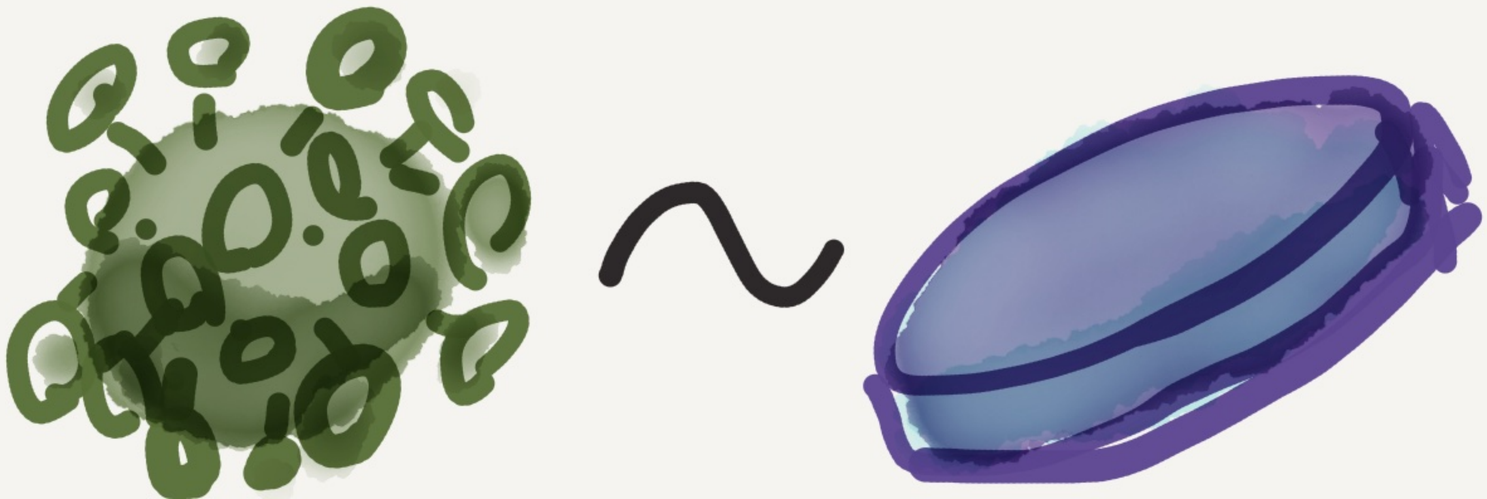
What's the median housing price in LONDON?



Which Serotypes should we VACCINATE against?



Is there a negative correlation between Truvada use and HIV transmission?



What's the total mass of all main-sequence stars?



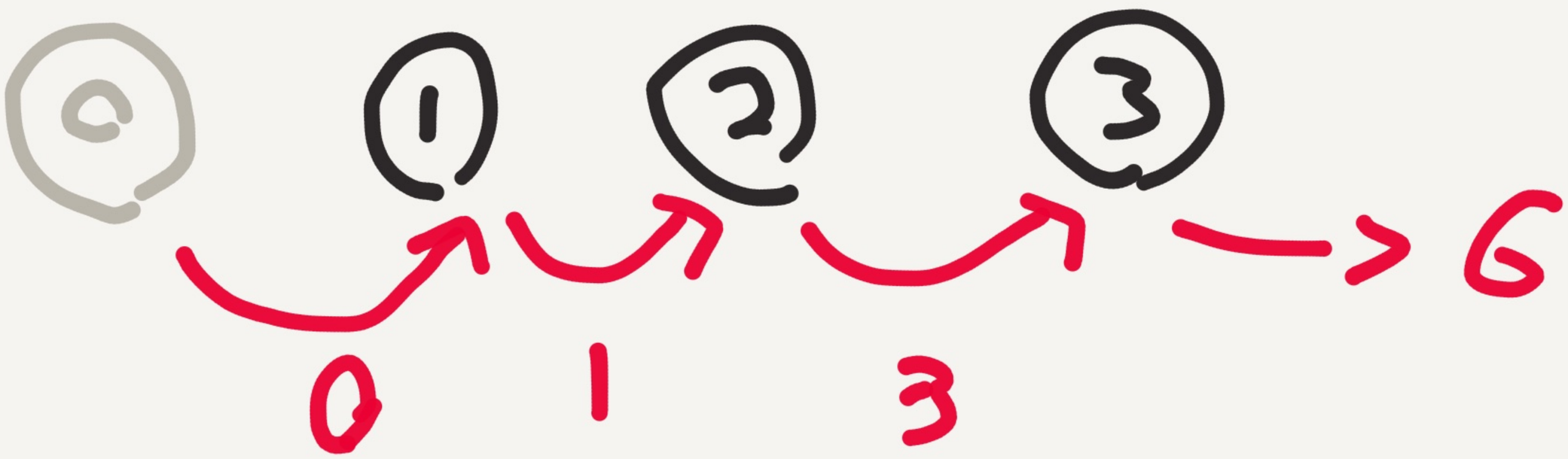
```
(->> stars
     (filter main?)
     (map :mass)
     (reduce + ))
```

hug

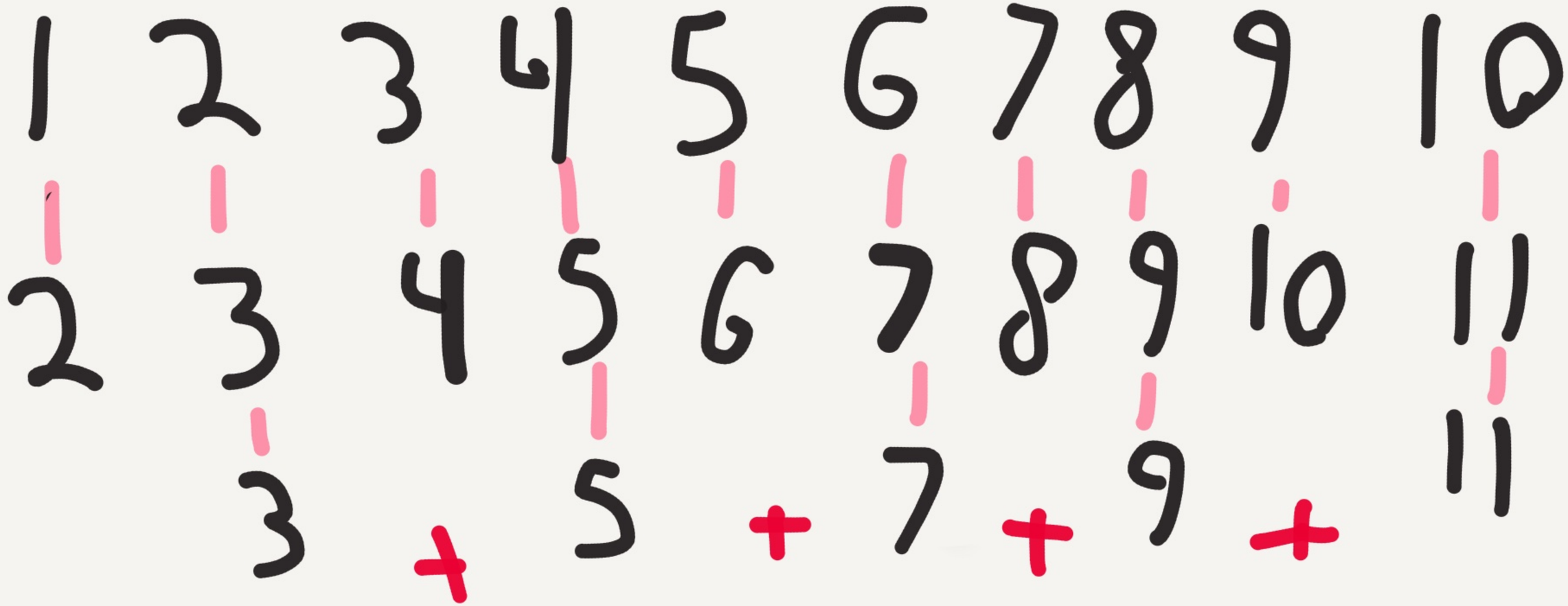
~~fold~~

(reduce +)

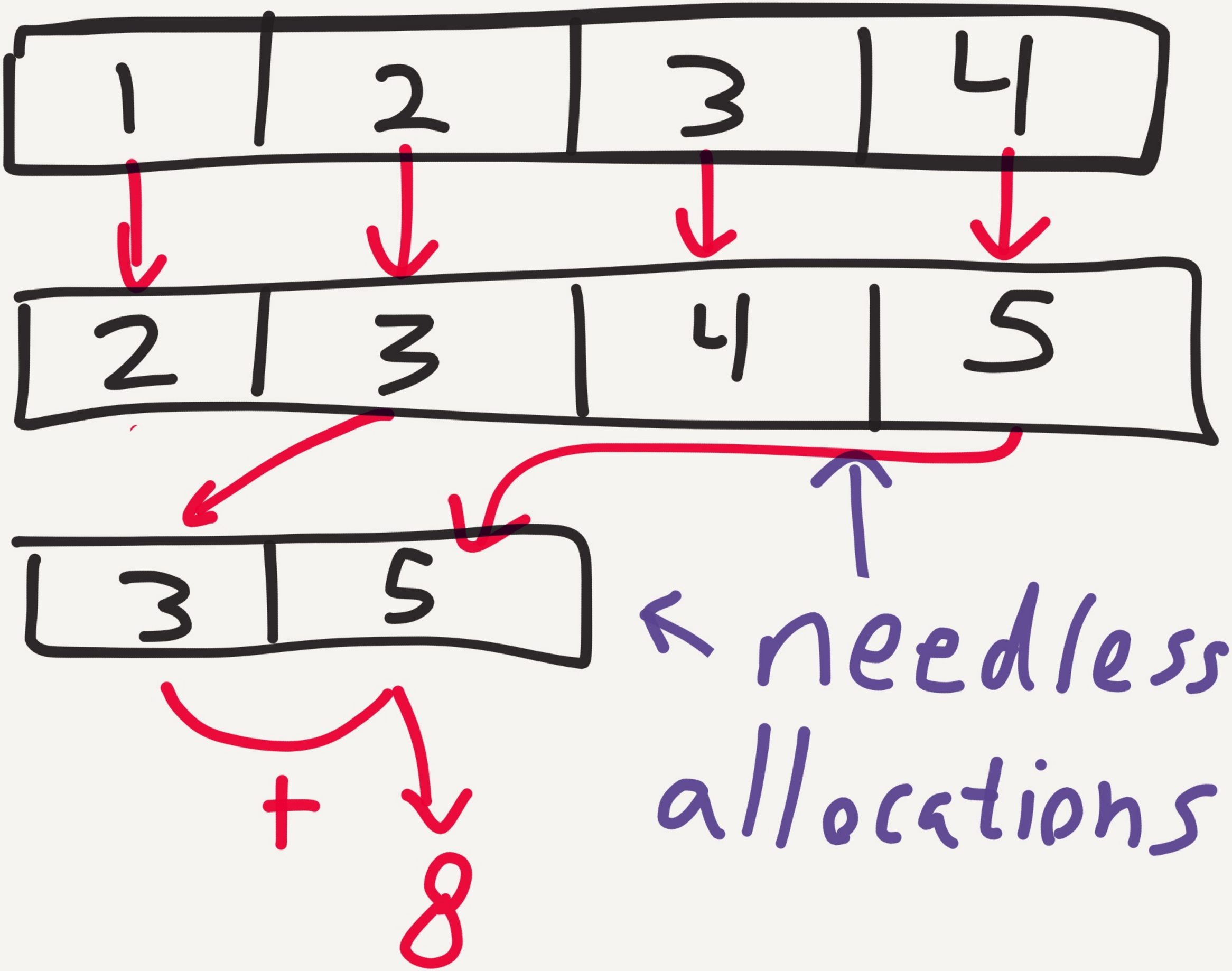
"reduce" (foldl)



(reduce + 0 [1 2 3])




```
(->> coll (map inc)
          (filter odd?)
          (reduce +))
```

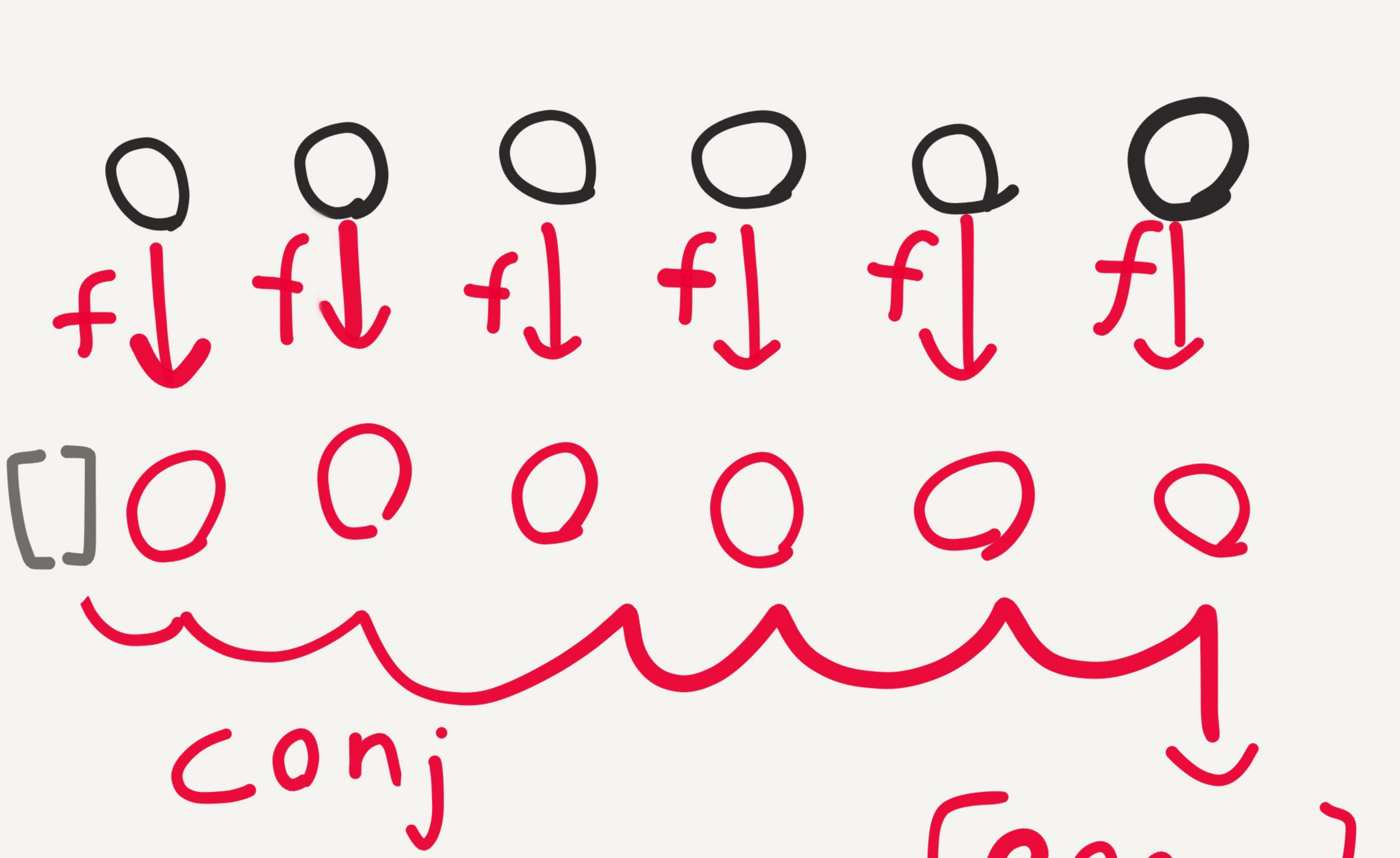

1
2
3
4
2
3
4
5
3
5
+
8
needless
allocations

(map) is a kind of

(reduce)



amorphism

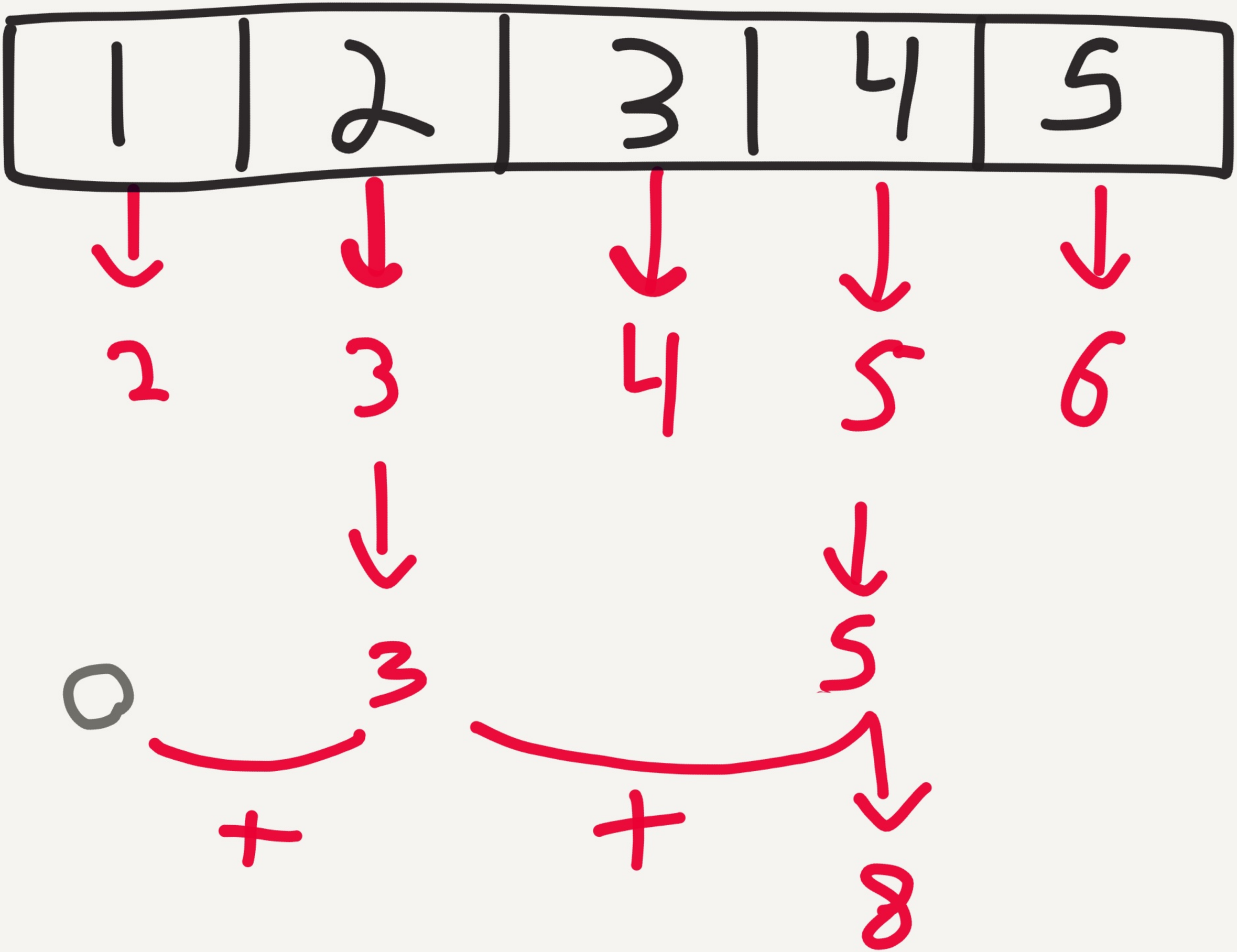

f
f
f
f
f
f
[]
conj
[ooo...]

filter, take,

keep, count...

all kinda monoids

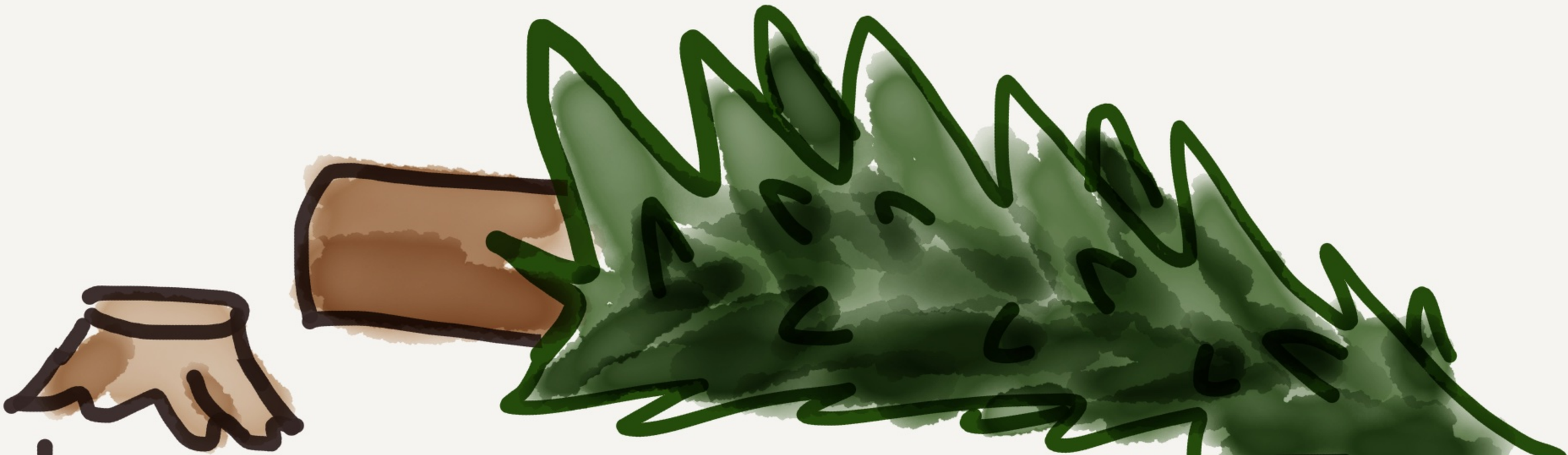
So: fold all the seq ops into a reduce


1
2
3
4
5
2
3
4
5
6
0
3
+
5
+
8

Reducers ∌
Transducers
skip extra
allocations!

Stream

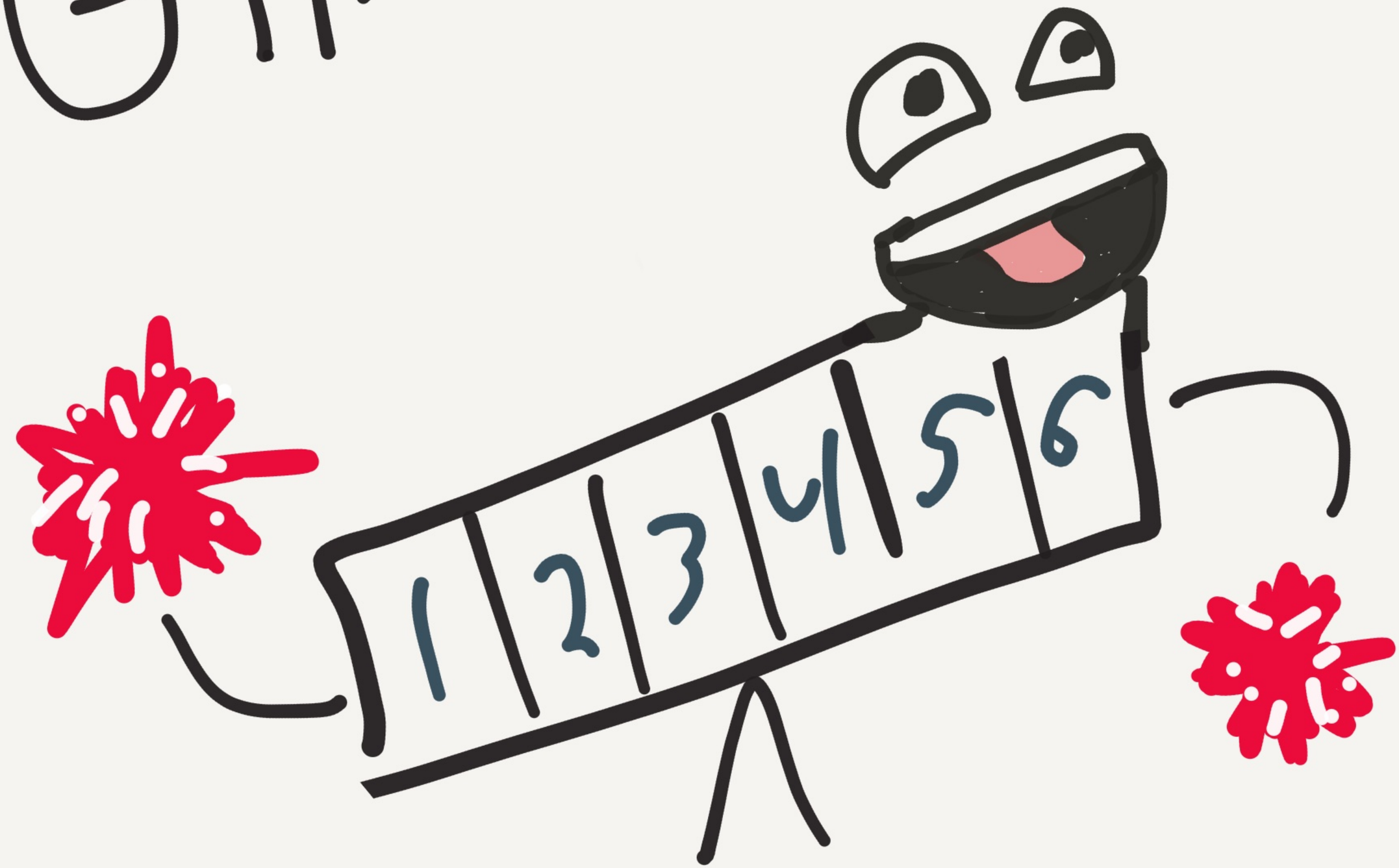
Fusion >=



Deforestation

Reducers
are defined
by each
Collection!

GIMME AN f!



(->> cheers

(r/filter fun?)

(reduce str))

```
(->> ??? (filter fun?)
     (reduce str))
```

Transducers have NO Underlying Type

(map inc)

"Over what?"

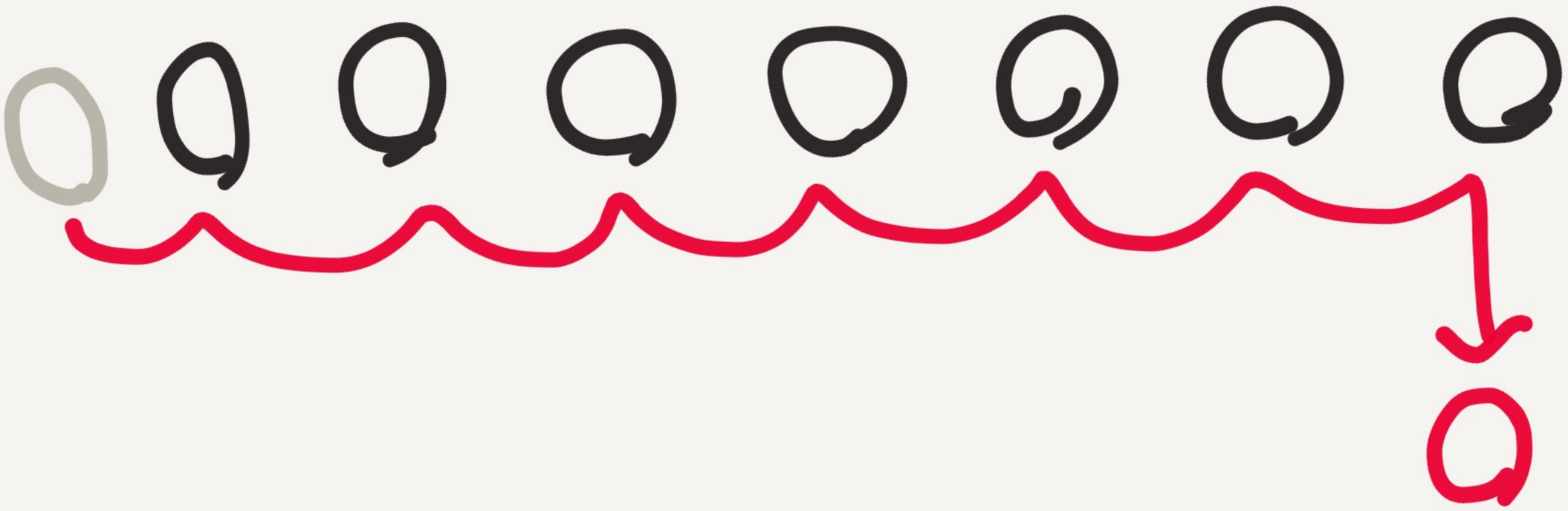
"ANYTHING!"

Transduce over channels OR segs!



But transducers are still Sequential

reducers



transducers



Don't you wish there were a Better Way™?

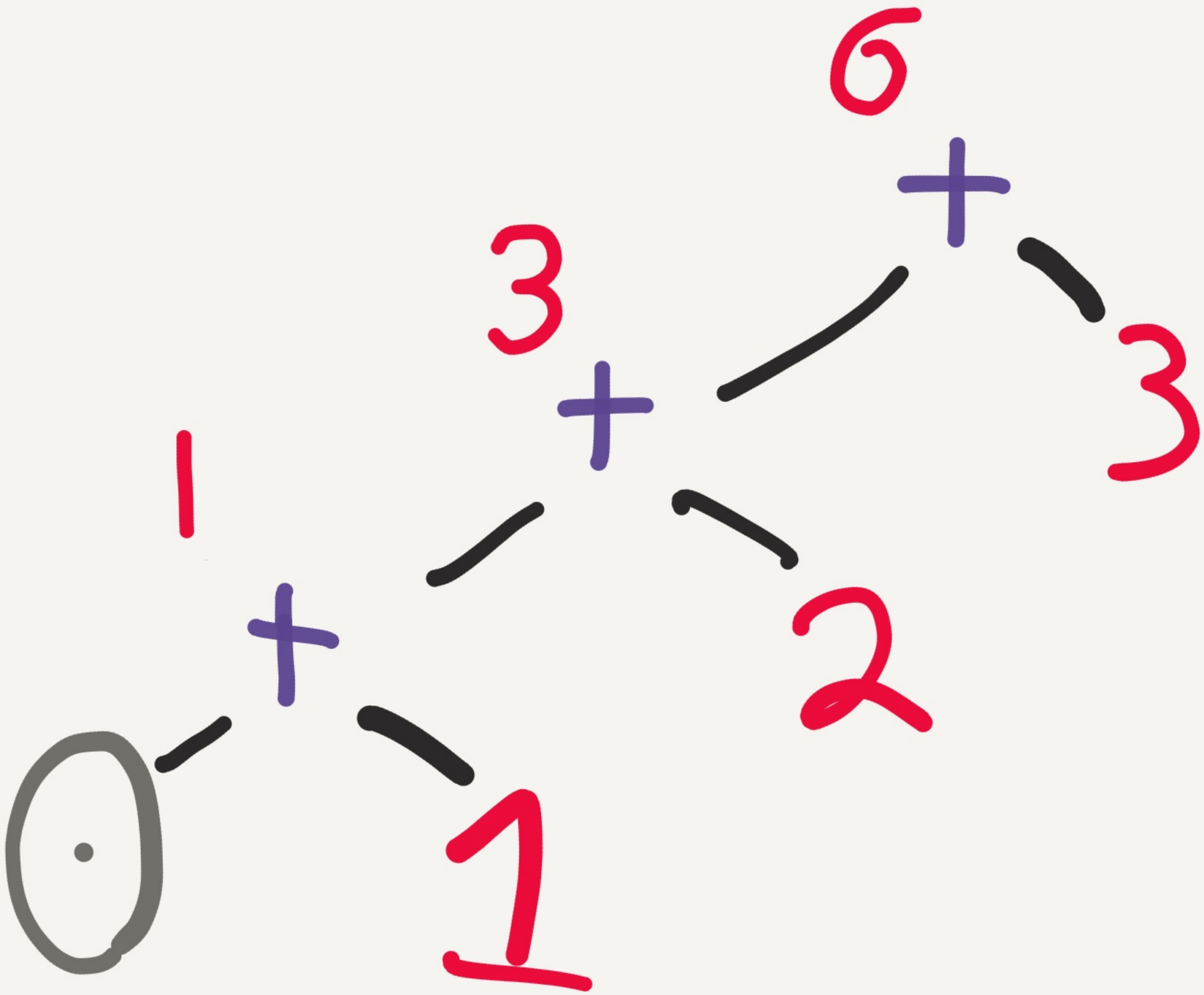




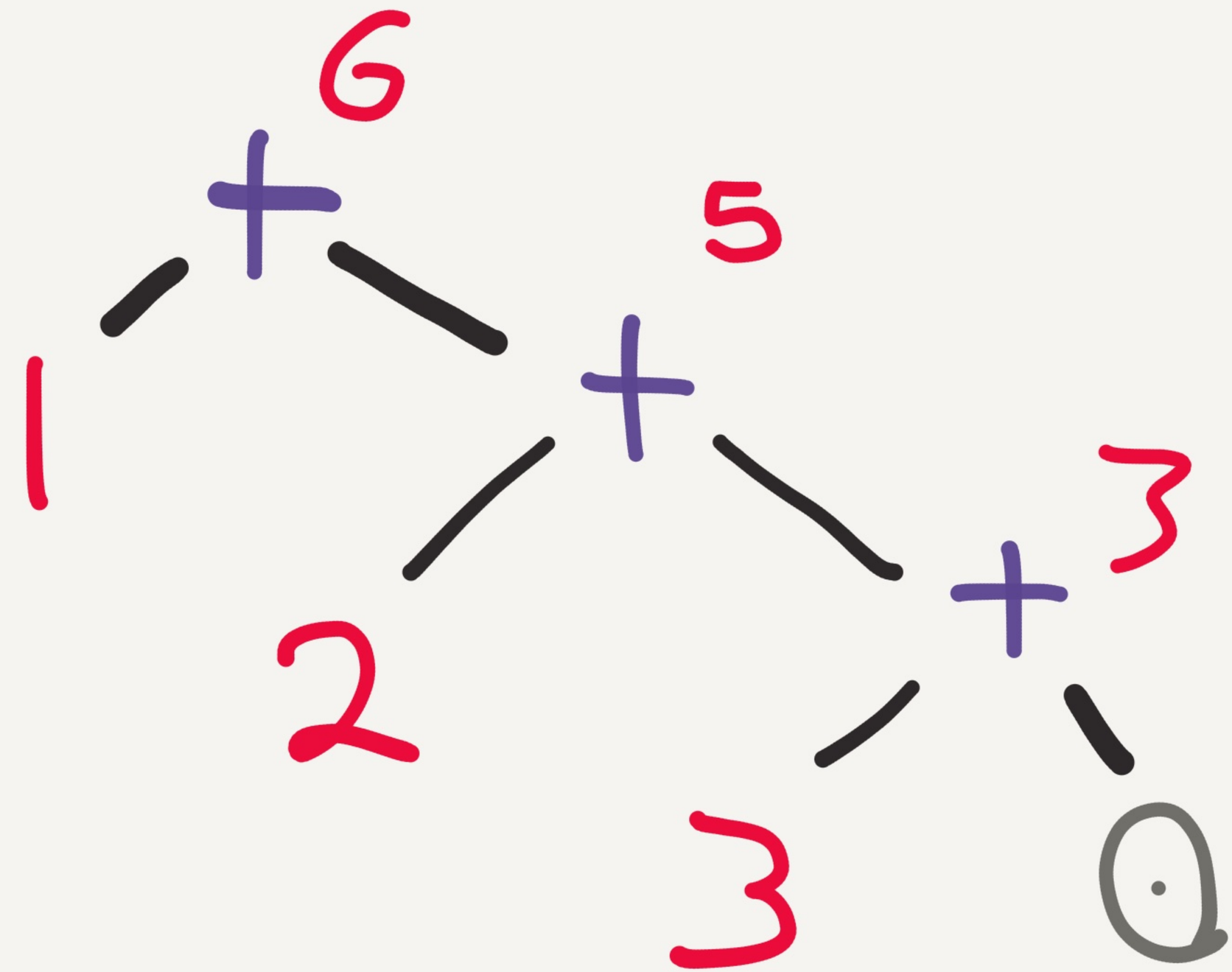


A WRINKLE - in - Time

(fold(+ 0 [1 2 3]))



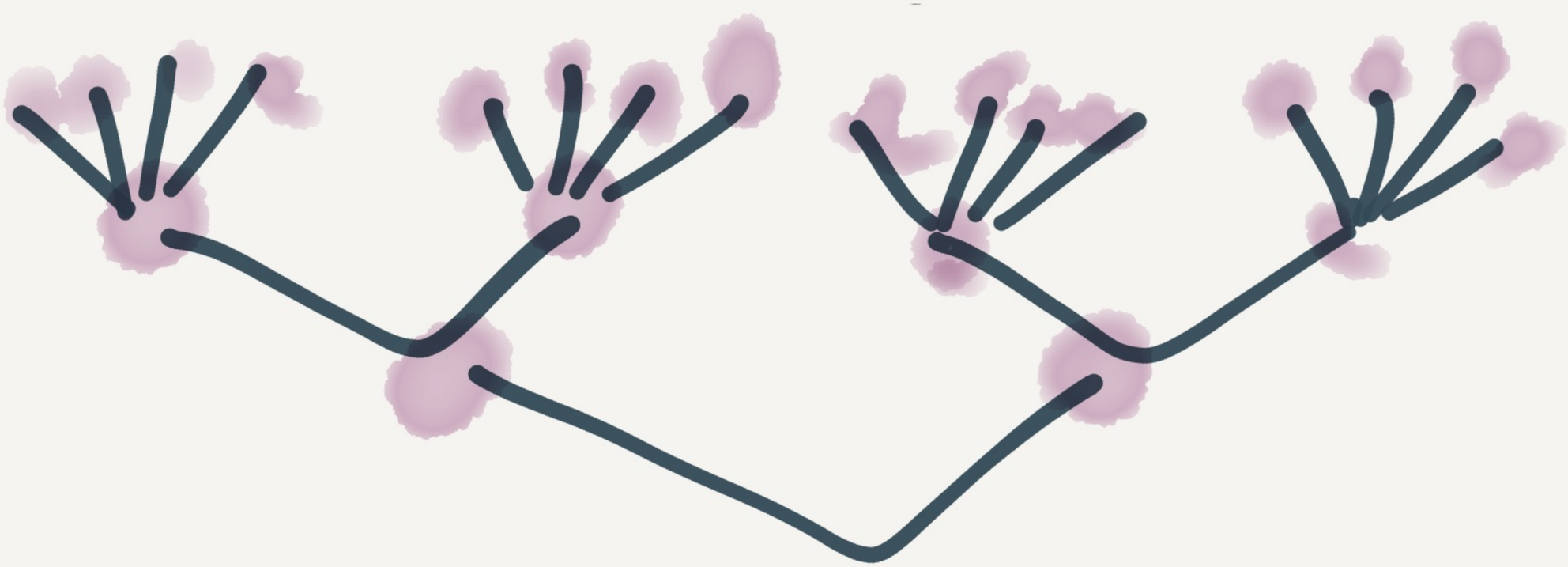
(foldr + 0 [1 2 3])

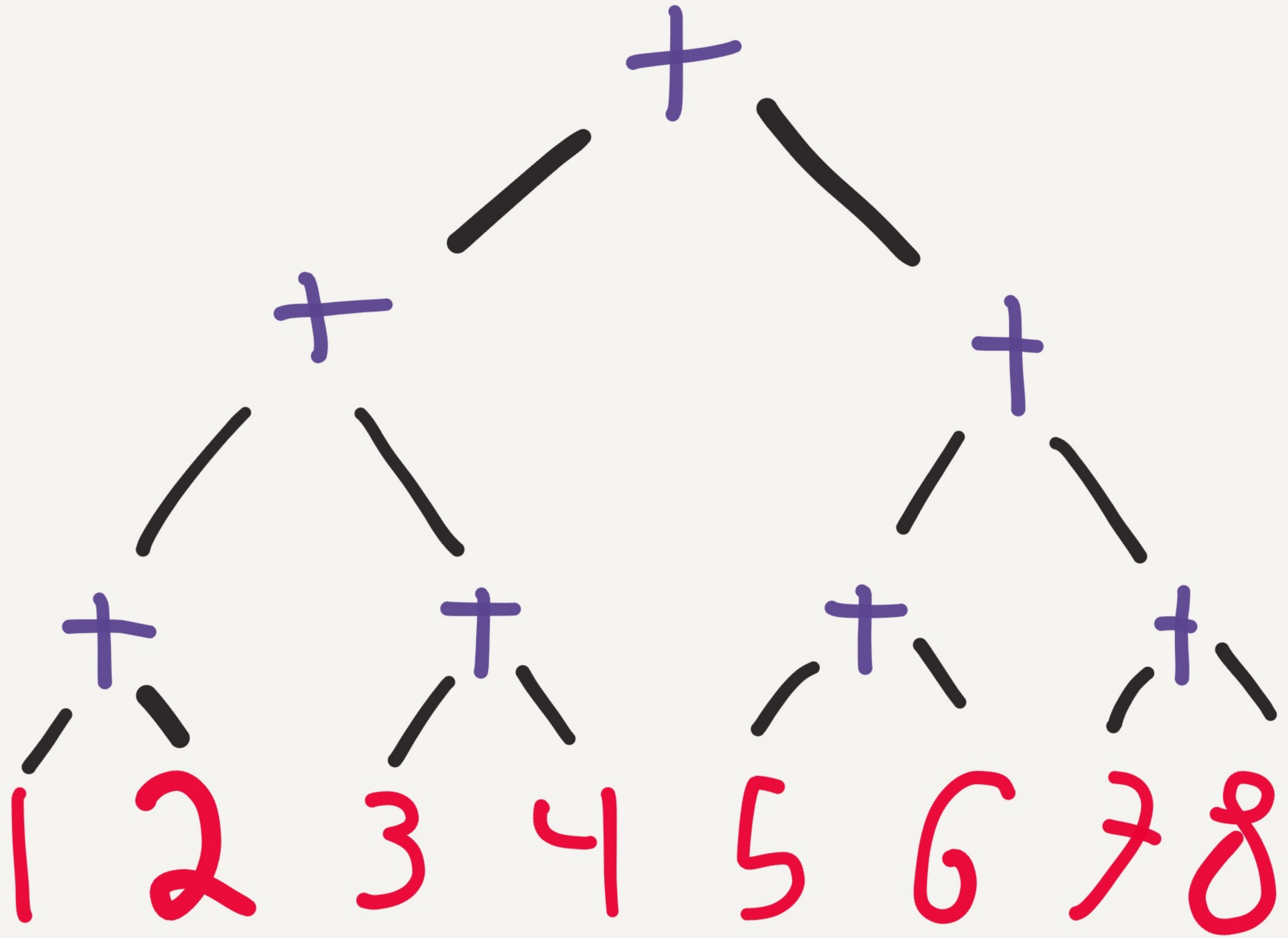


If f is ASSOCIATIVE:

$$(f\ a\ (f\ b\ c)) = (f\ (f\ a\ b)\ c)$$

Arborescent

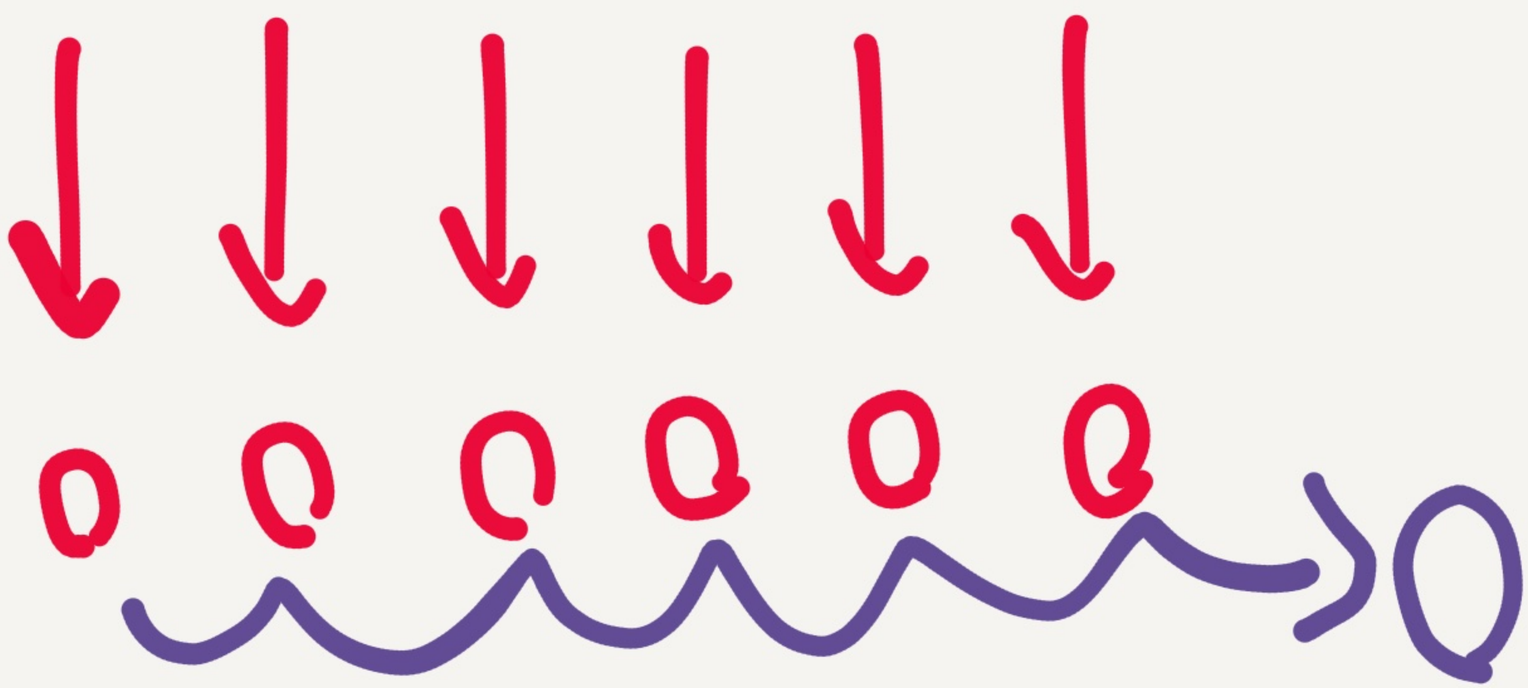



1 2 3 4 5 6 7 8

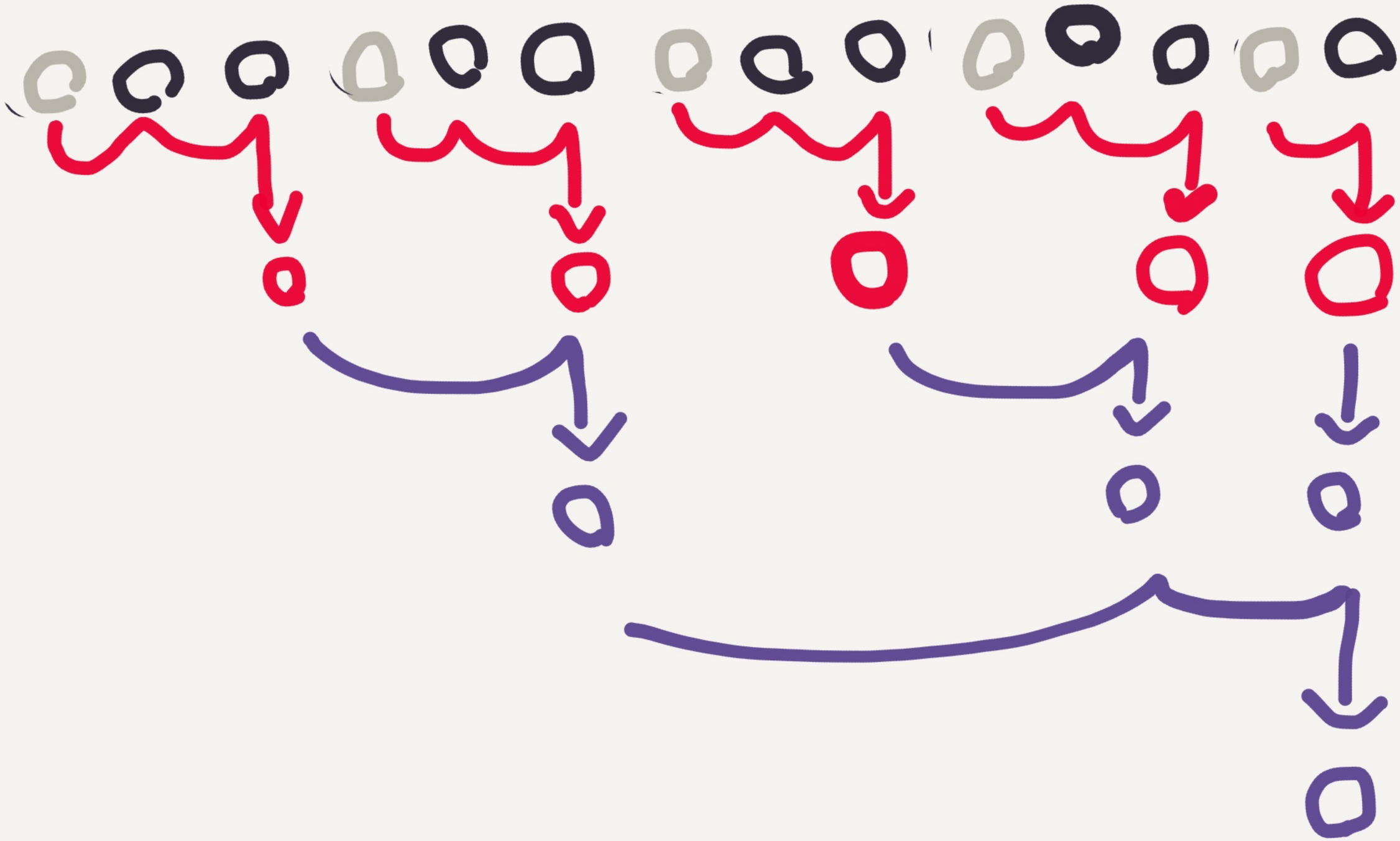
Independent chunks can be reduced in parallel



Then reduced values need to be combined

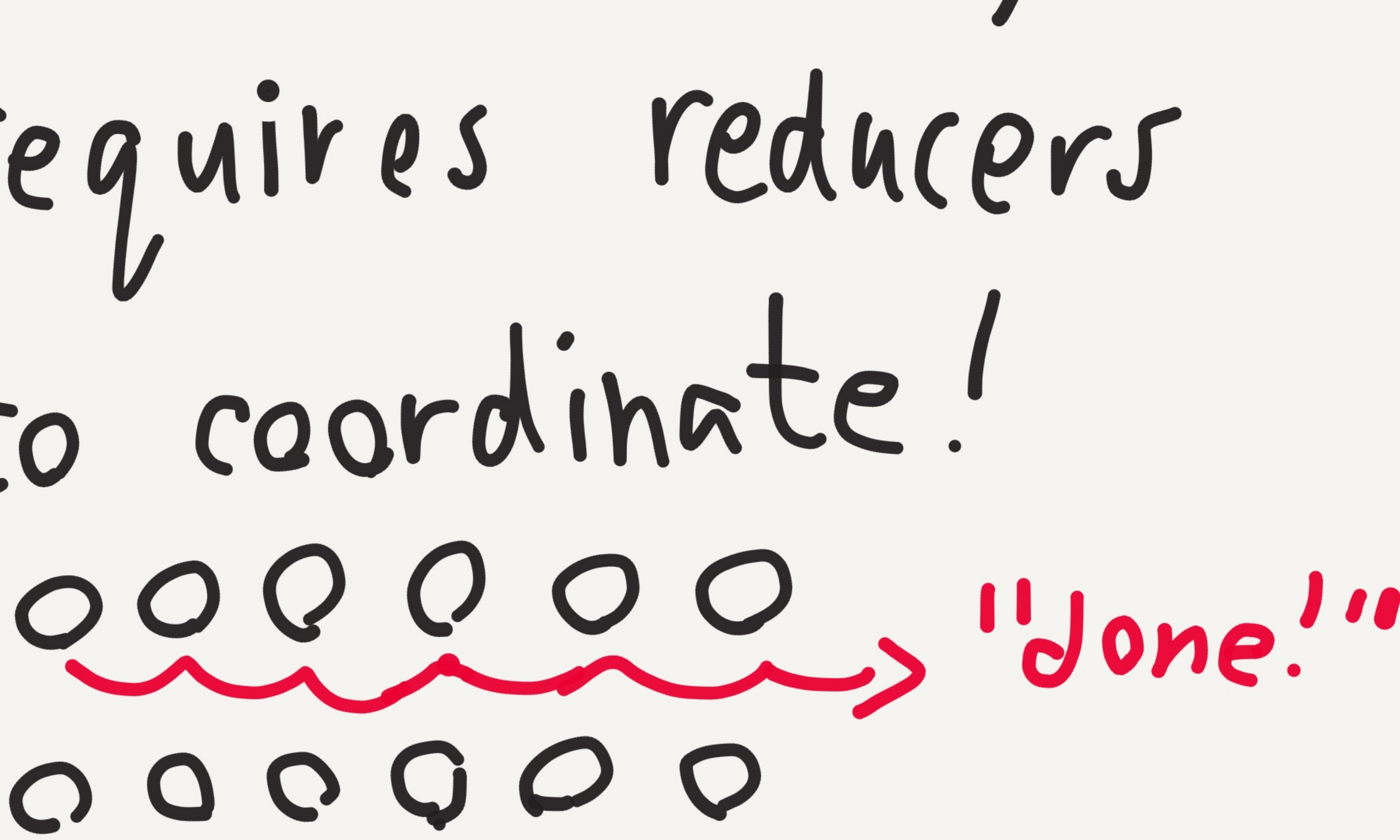


reducers fold (fj)

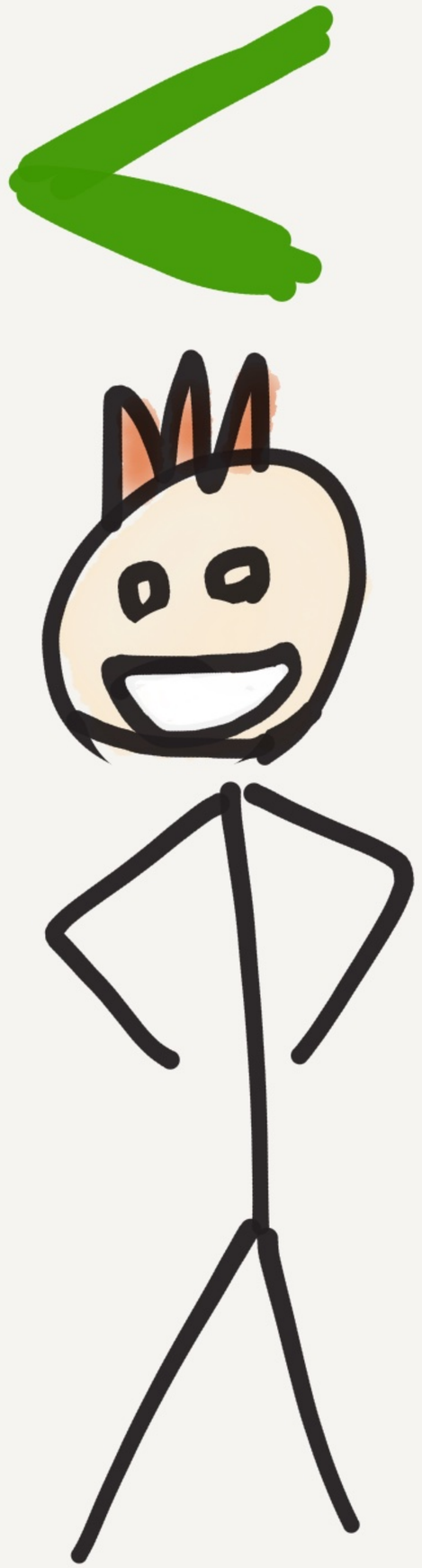


Independent reduction lets us go from A to N in $O(\frac{N}{P})$ time

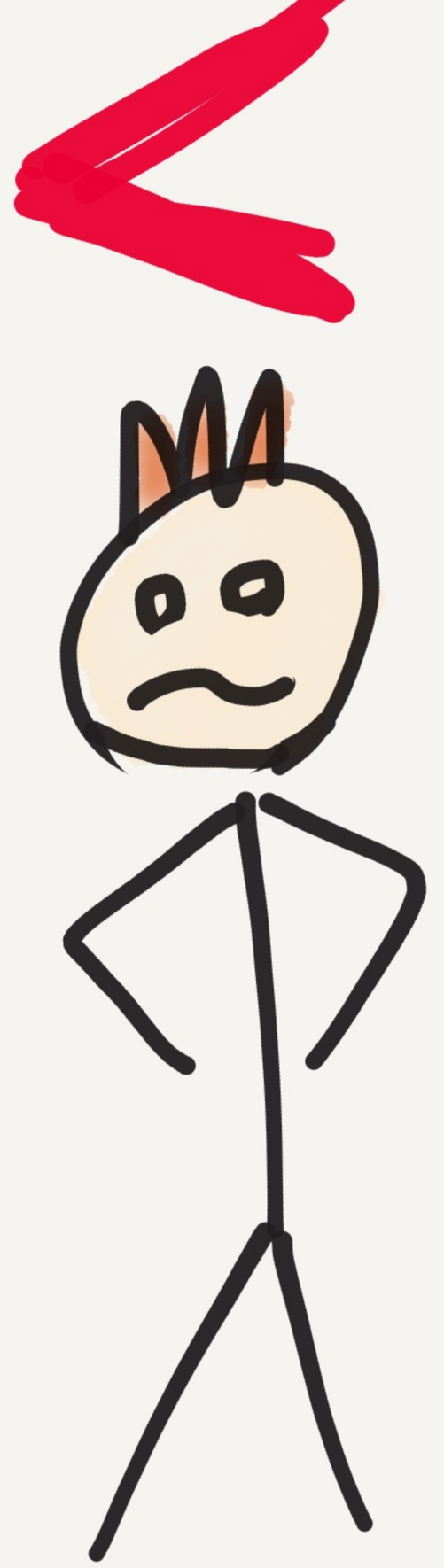
Even reducers/fold requires reducers to coordinate!



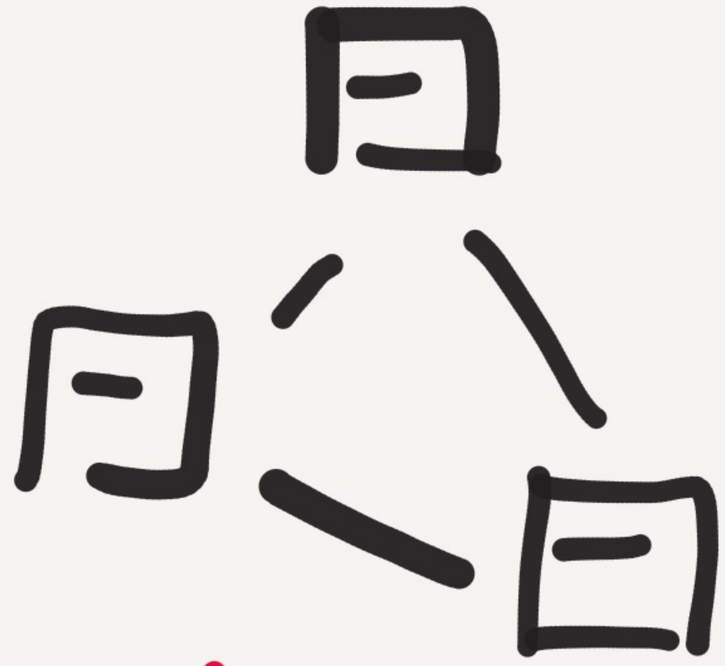
"Results will PRESERVE order"



Order requires COORD-INATION

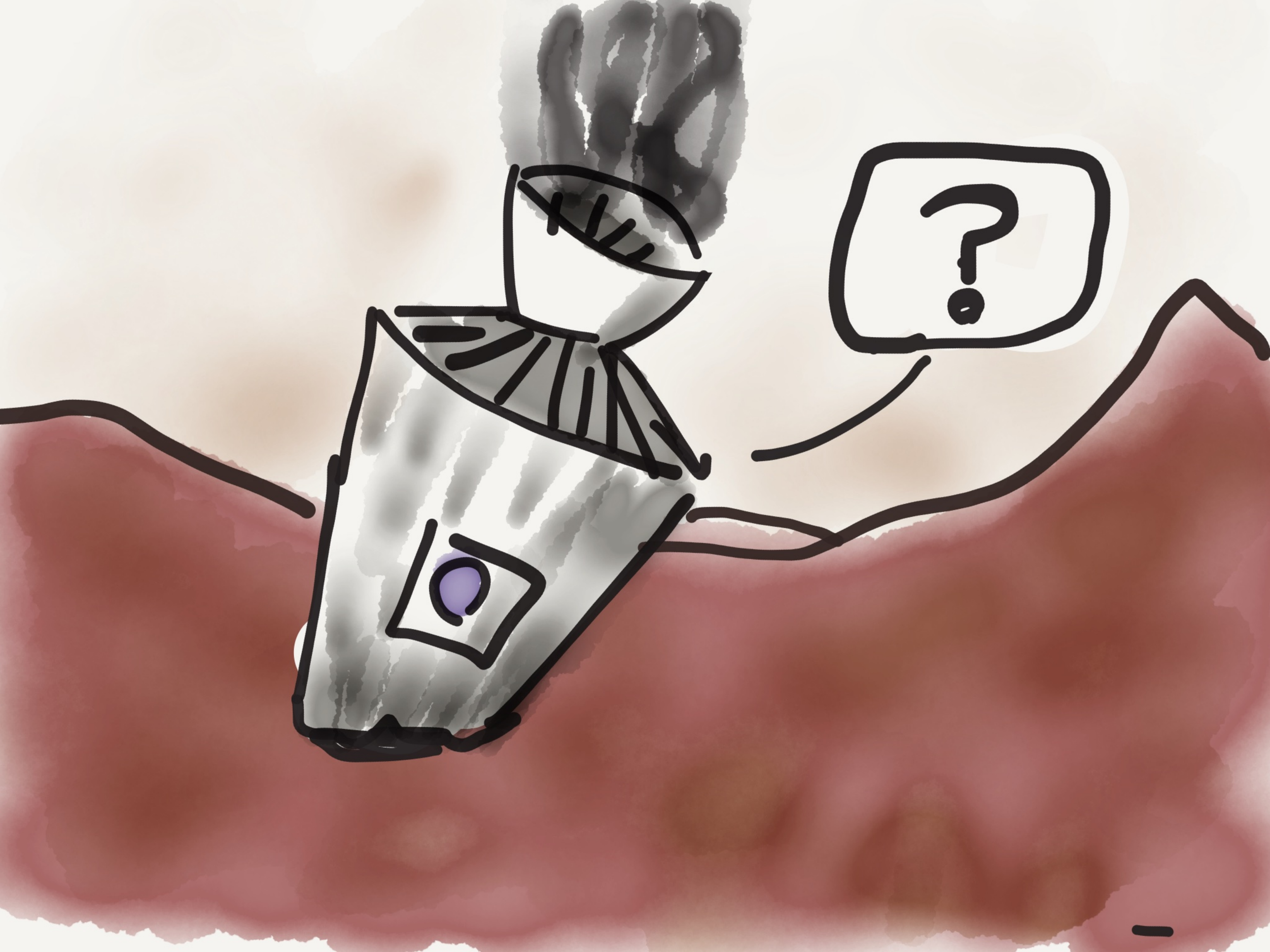


What if



Coordination

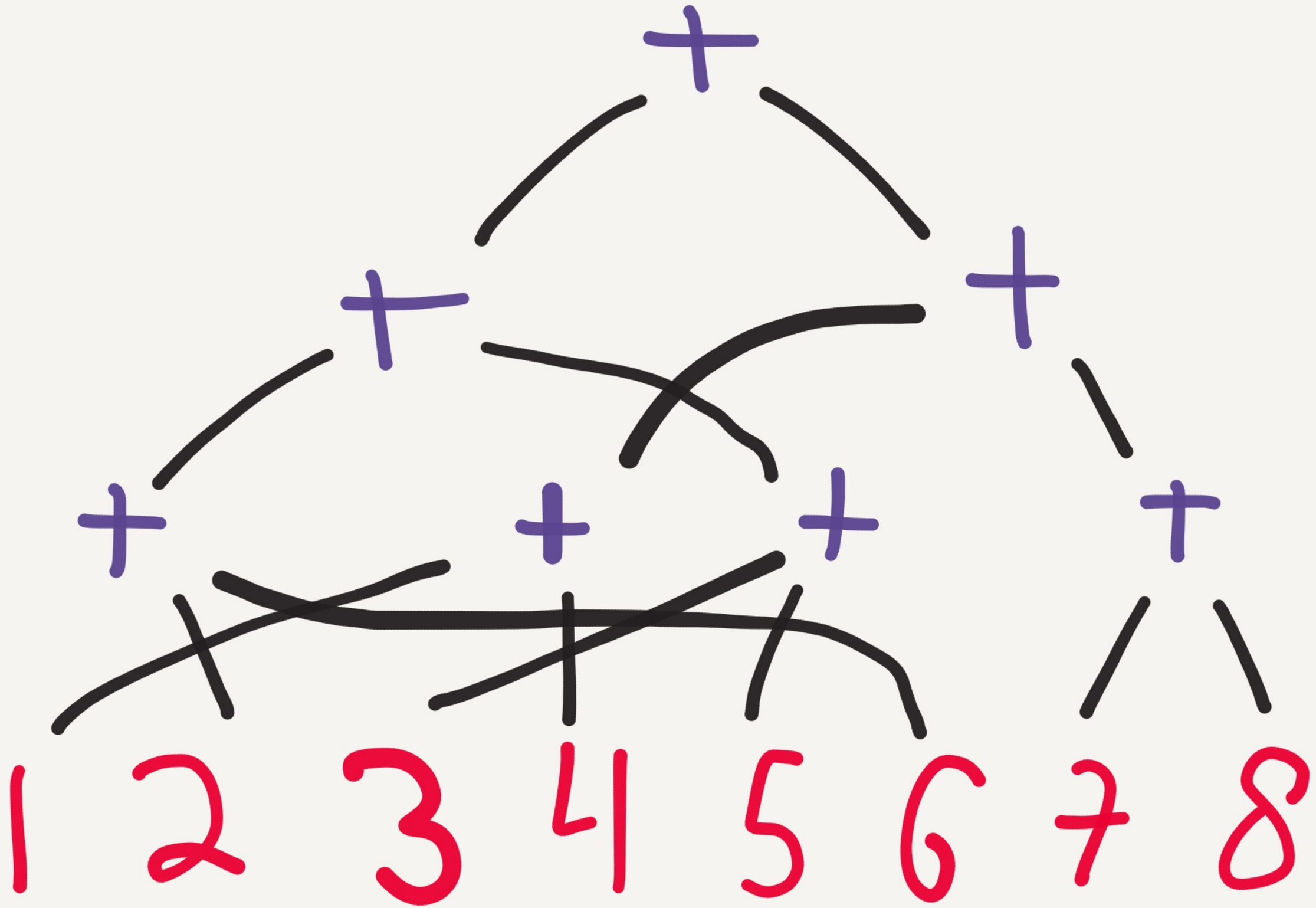
is HARD?



If f is

COMMUTATIVE:

$$(f\ a\ b) = (f\ b\ a)$$


1 2 3 4 5 6 7 8

A function that's

- Associative

- Commutative

And an identity...


COMMUTATIVE
MONOIDS
f(x, f(y, z))

Monoids Compose Wonderfully

Monoids which COMMUTE are GREAT for distributed systems

See Also:

CRDTs

(different talk)

TESSER:

- Parallel
- Unordered
- Stream fusion
- Collection independent

Same great

SEQ

API!

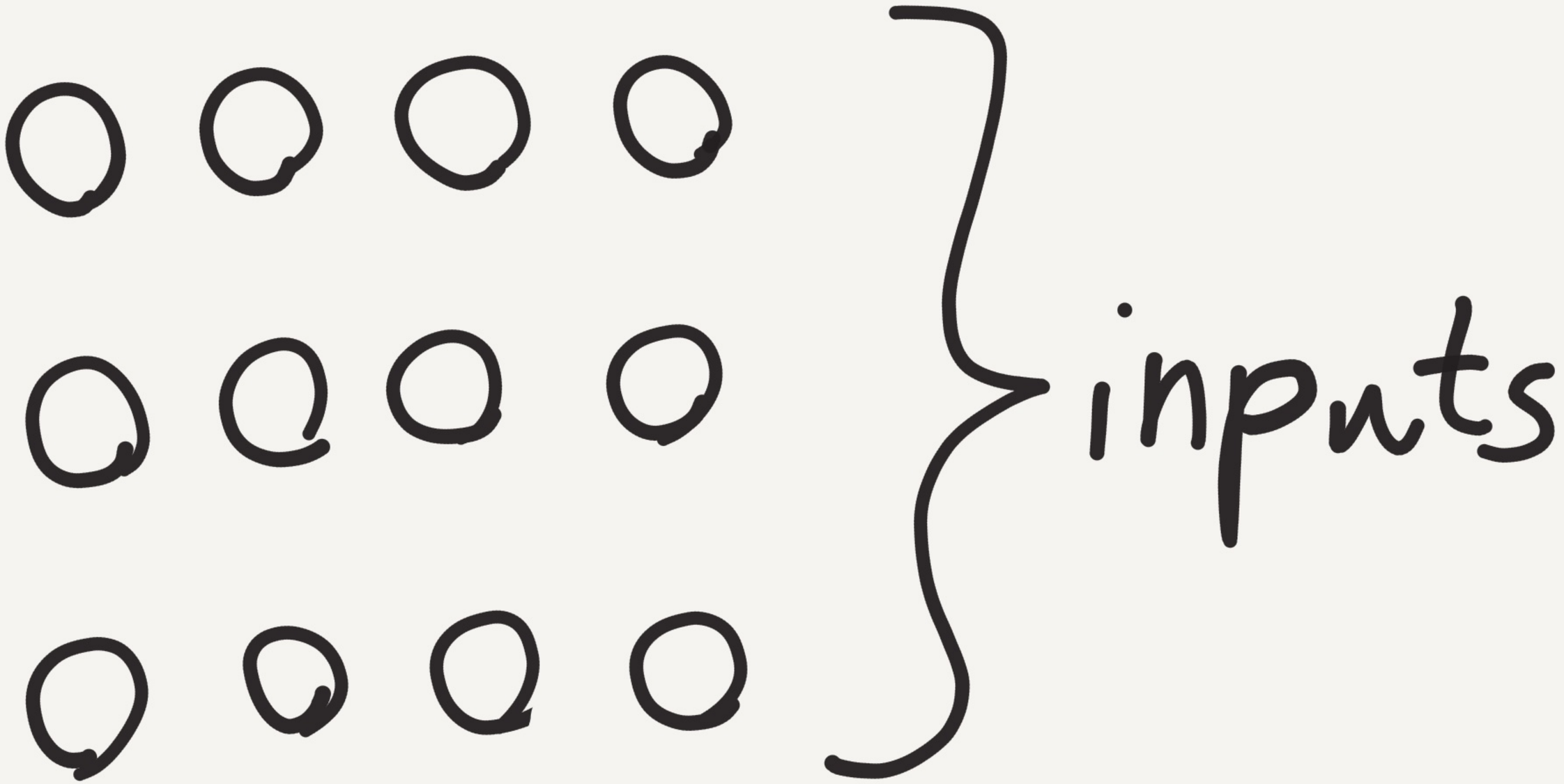
```
(->> (t/map inc)
     (t/filter odd?)
     (t/reduce + 0)
     (t/tesser colls))
```

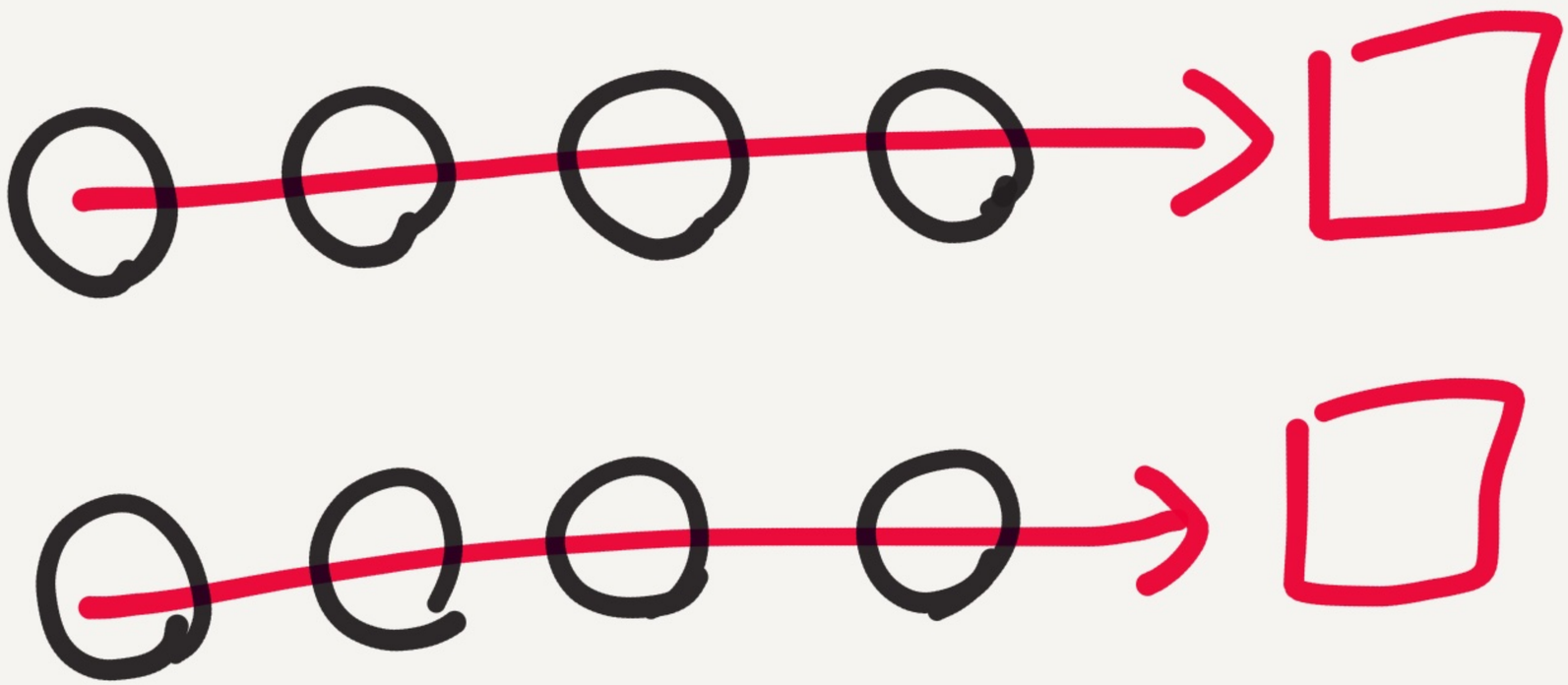
Can run on
many threads
locally...

Or on Hadoop using stateful mappers!

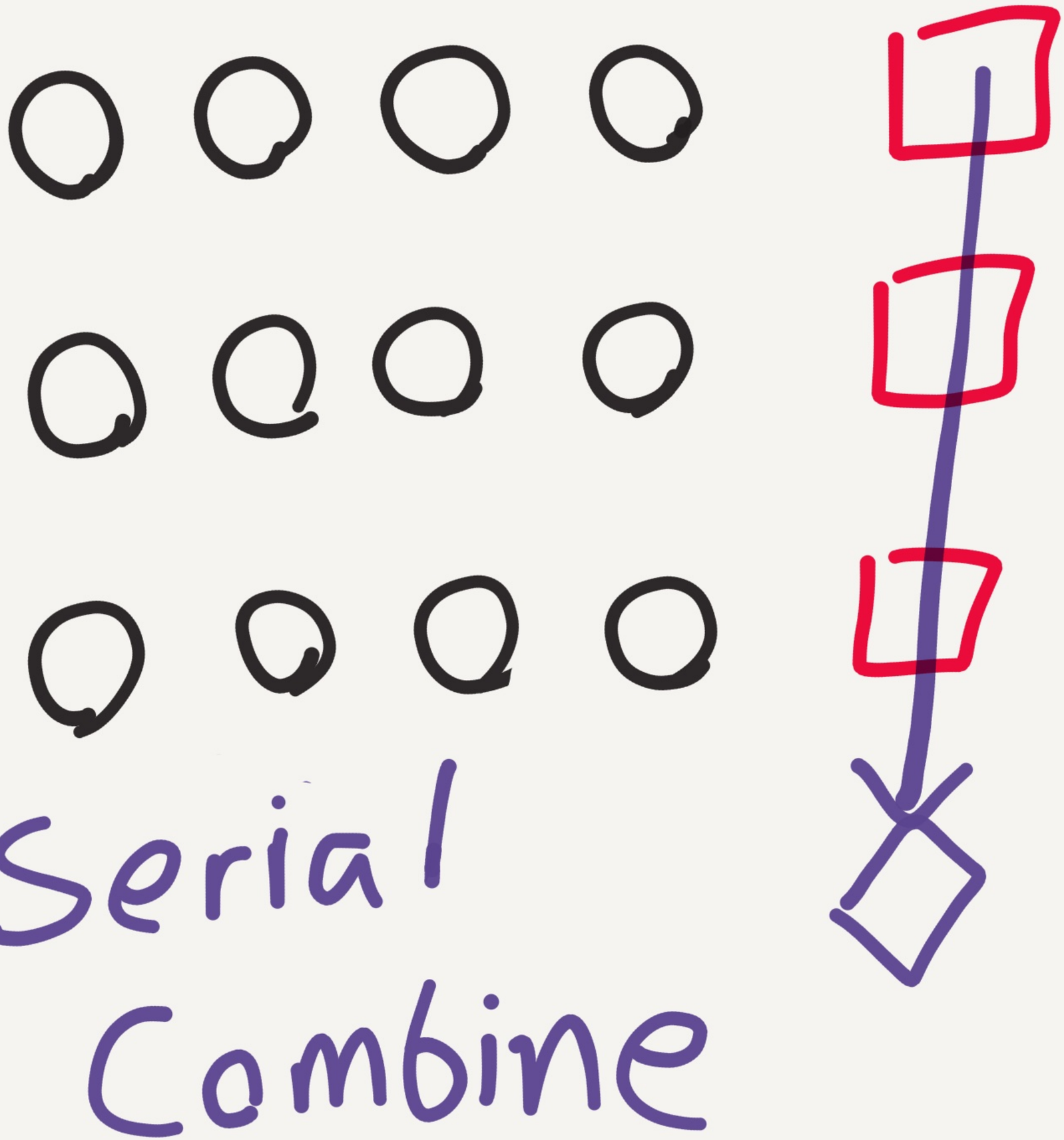


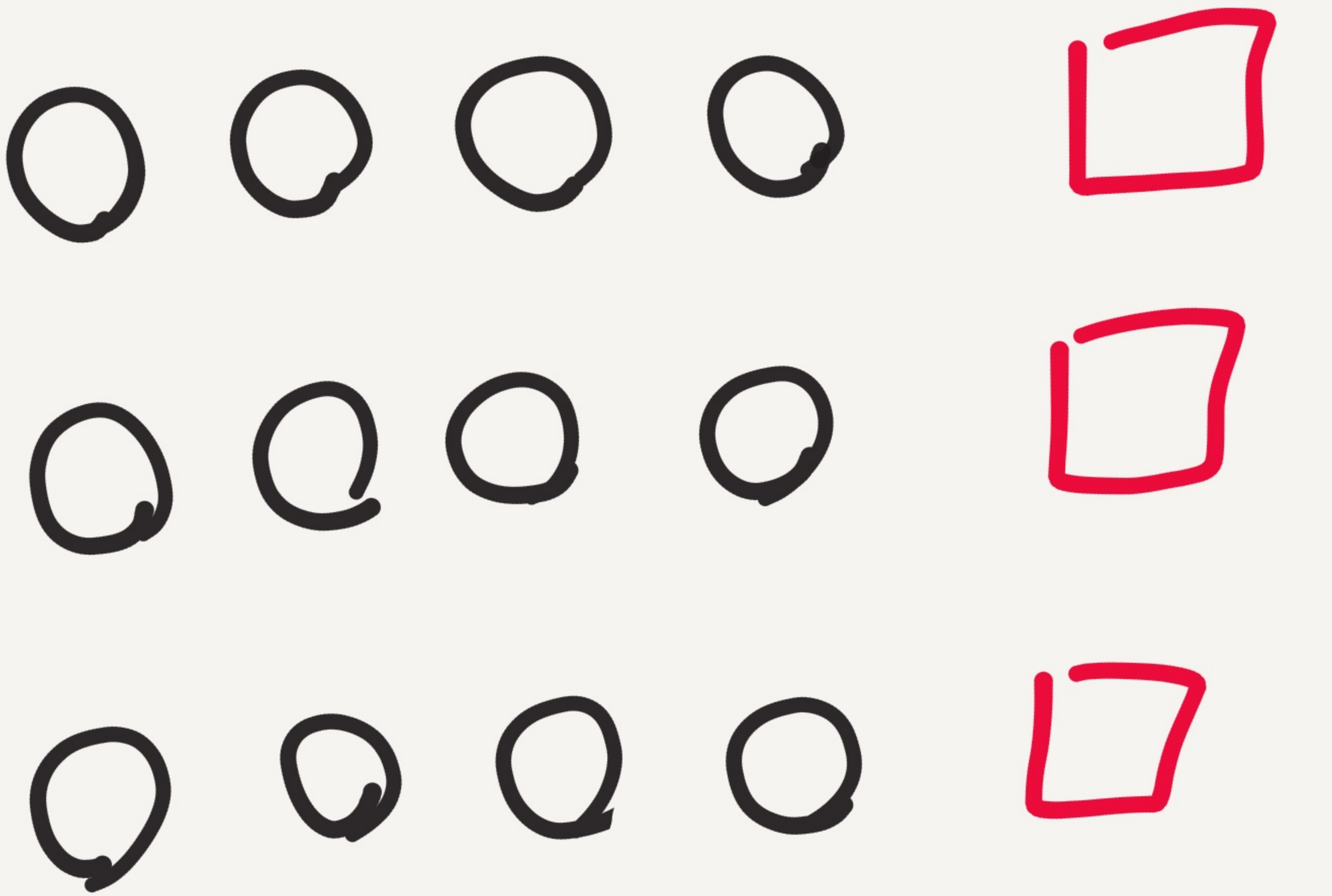
How?


inputs



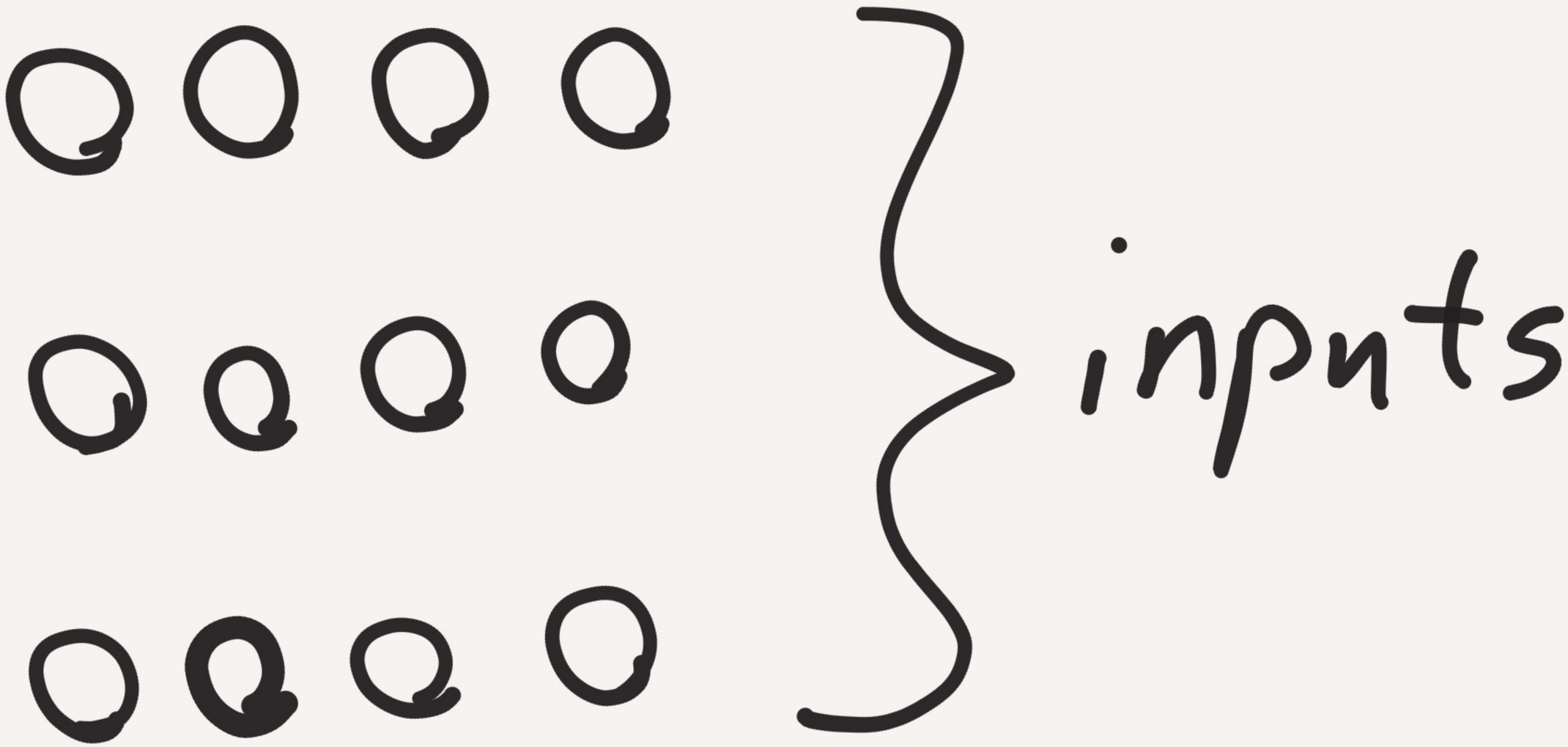
Concurrent reduce


Serial
Combine


result



Complection made Easy!

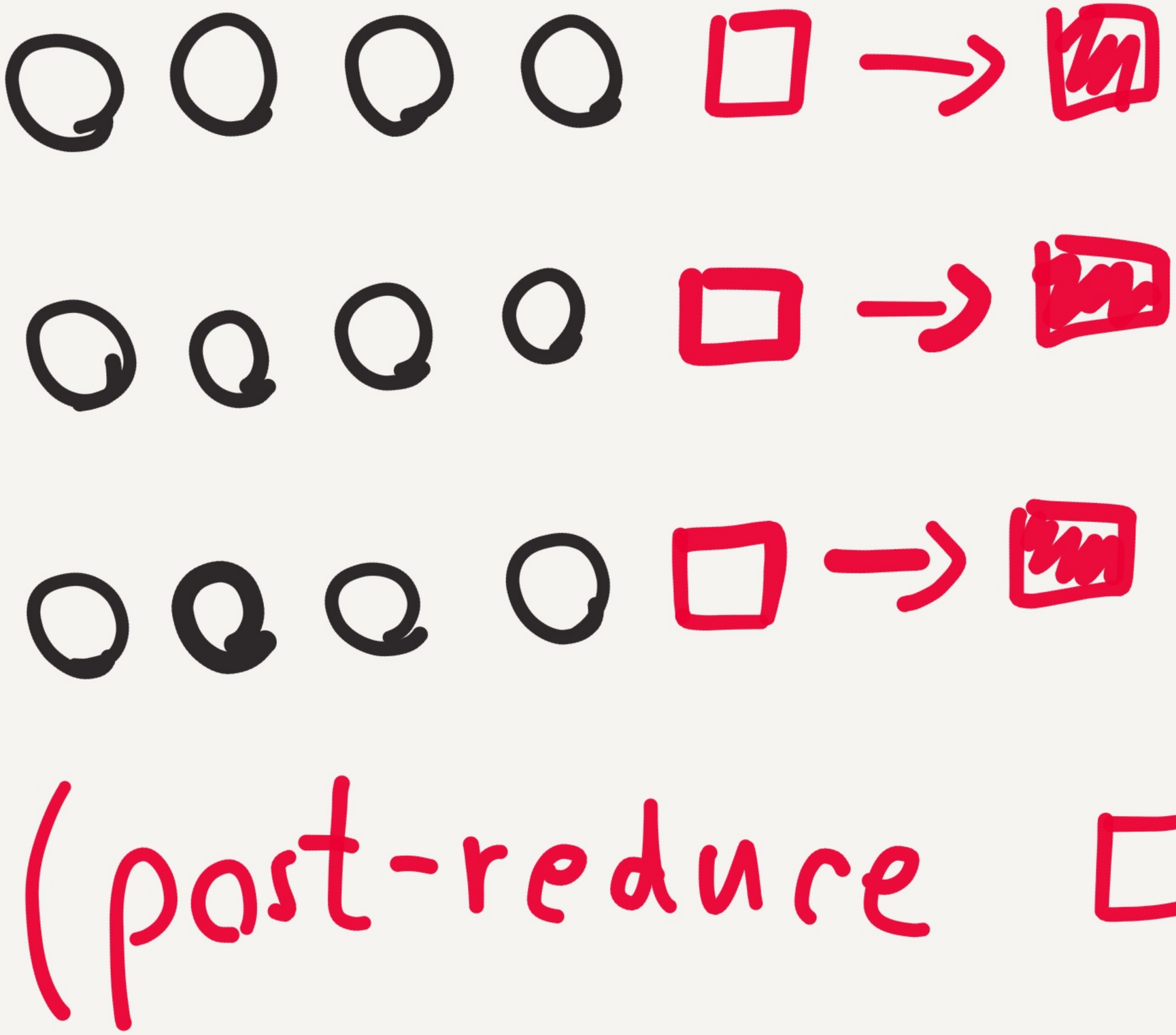

inputs

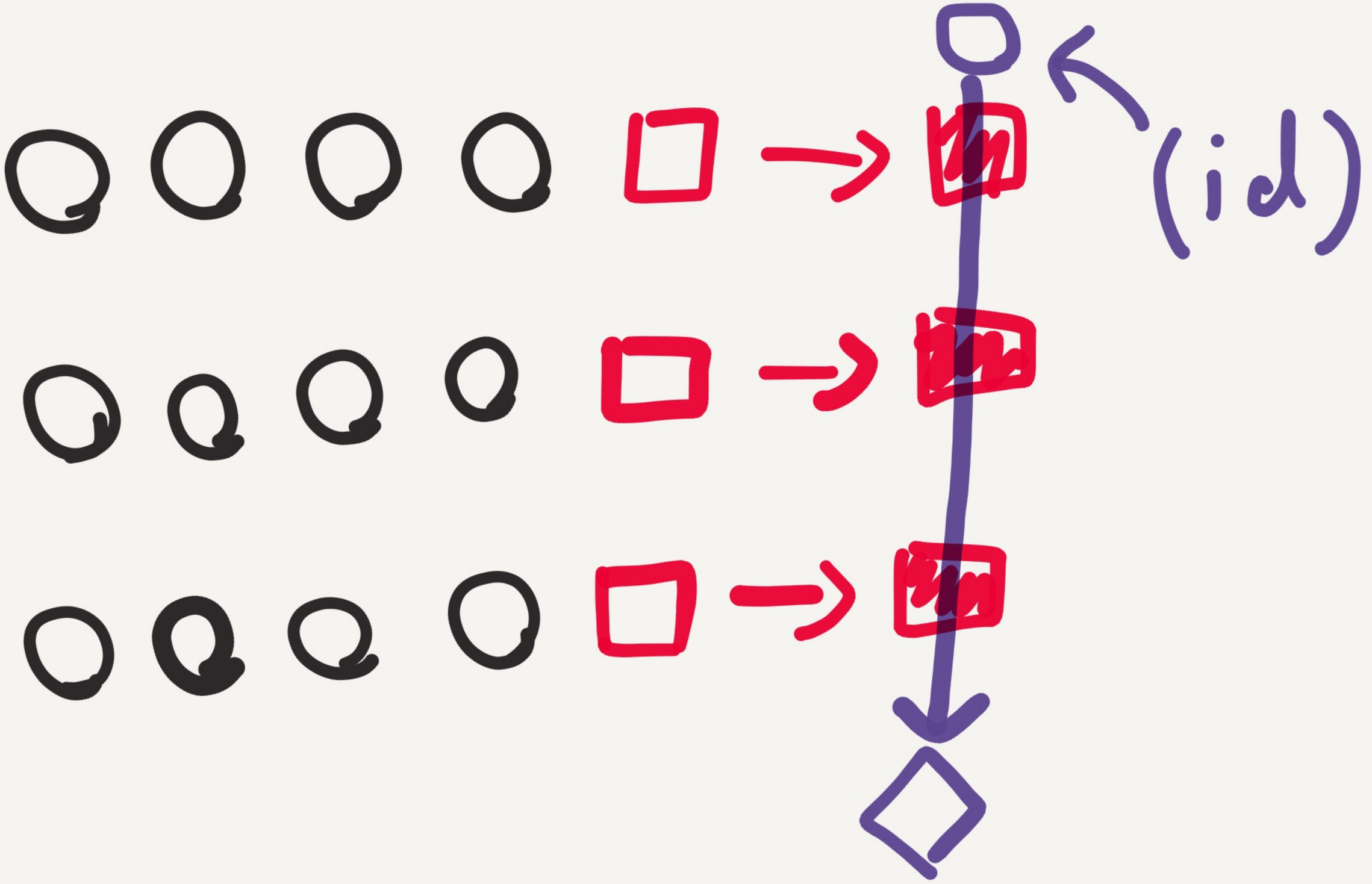


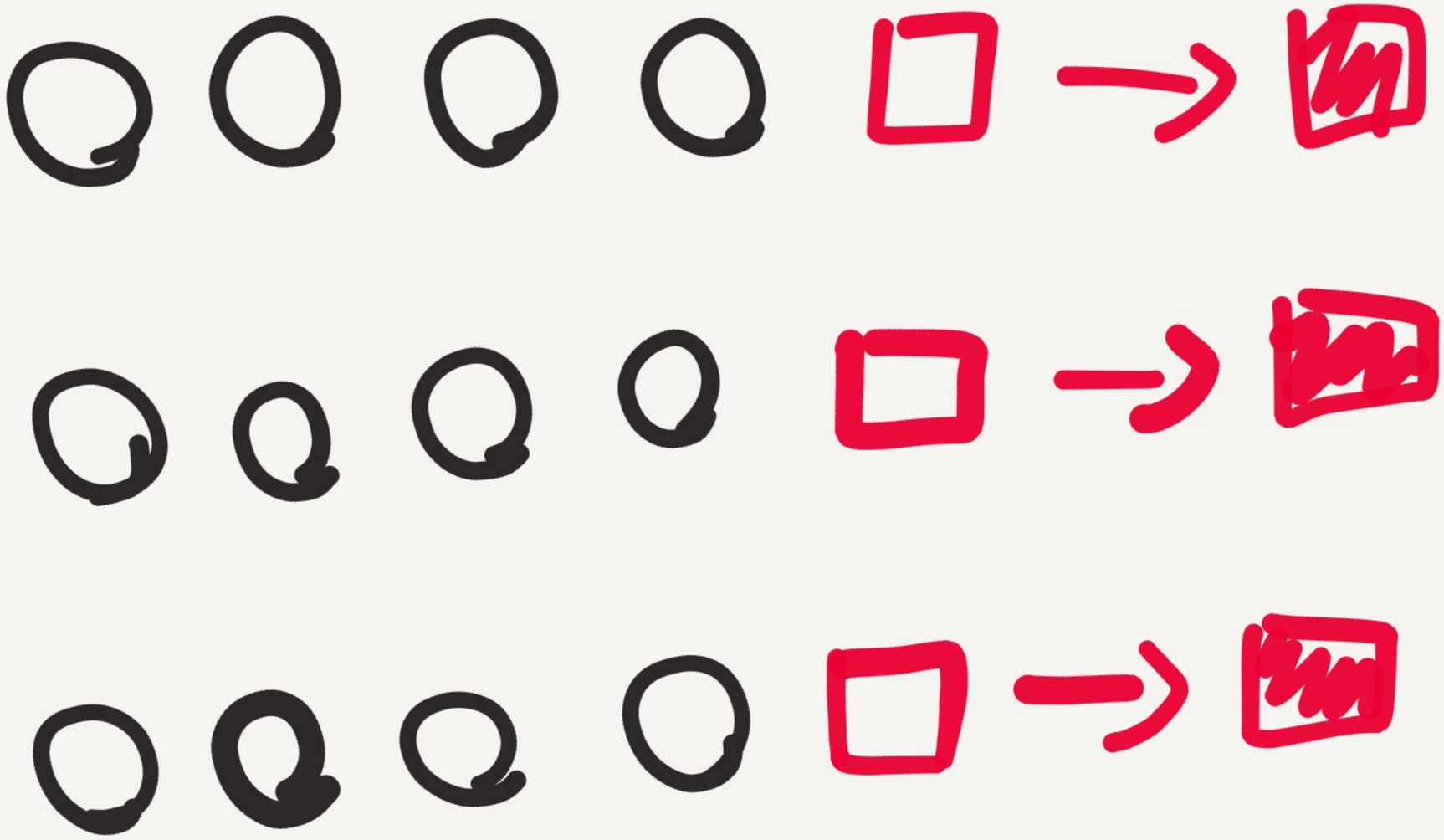




(identity-fn)


(post-reduce □)


(id)

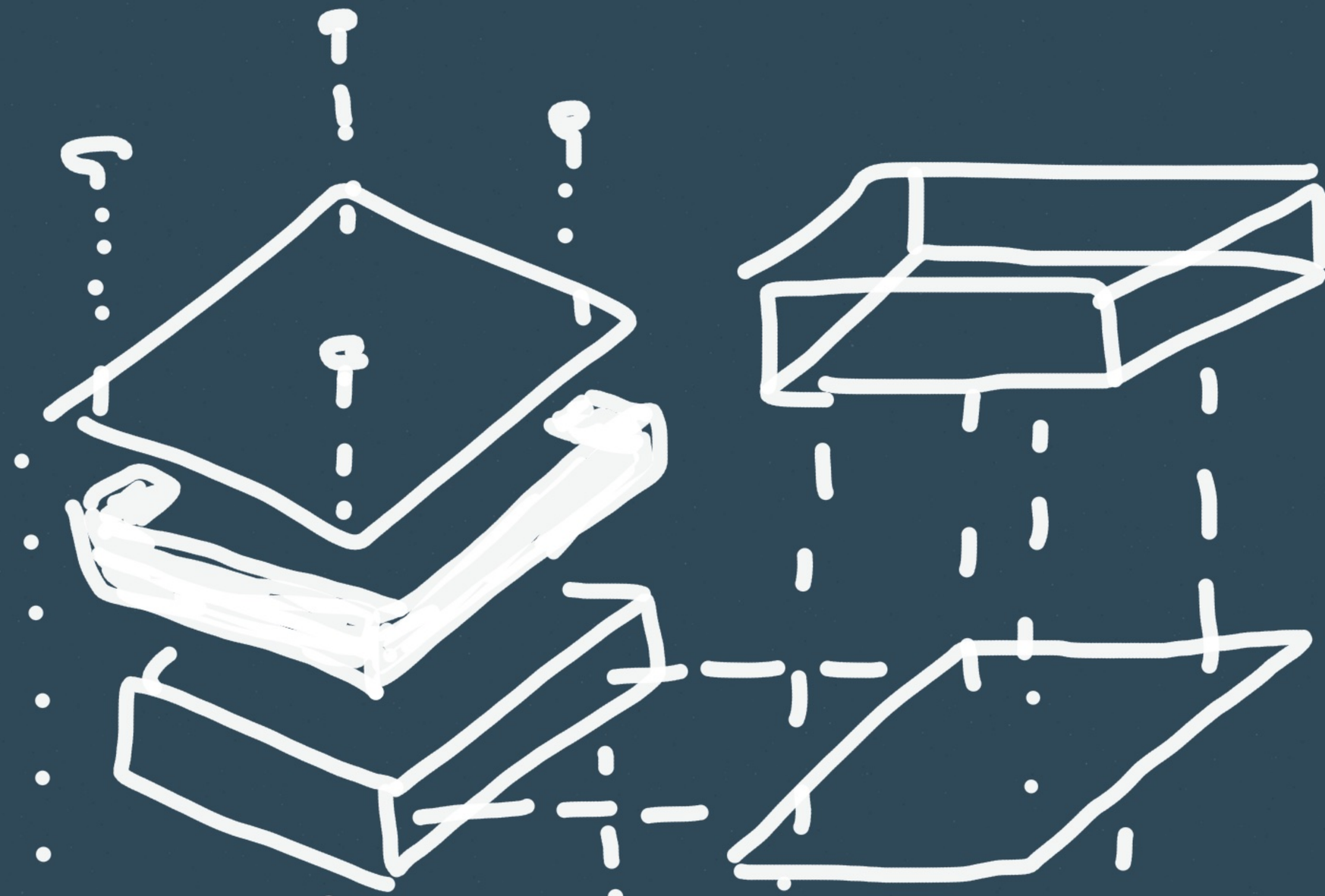



(post-combine)



result:





internals

Transducers:

```
(fn ([] ...)
    ([x] ...)
    ([acc x] ...))
```

identity

post

reduce

Reducers: (f) (f a x)

(r/fold reducef
combinef)

(f) (f a x)

Tesser uses a Map!

{
:reducer-identity
:reducer
:post-reducer
:combiner-identity
:combiner
:post-combiner
}

Functions like (map)
yield builder fns that
generate Tesser maps.

(Like Transducers!)

(map str)

↓

(fn [downstream]

{:identity (fn [] ...)

...

...})

Unlike transducers,
we defer the build
phase so we can
compose inside $out.

=> allows seq API

```
[(fn [down] ...)
 (fn [down] ...)
 (fn [down] ...)]
```

```
{:identity
 :reducer
 ...}
```



Constraints

Identity fns must be pure



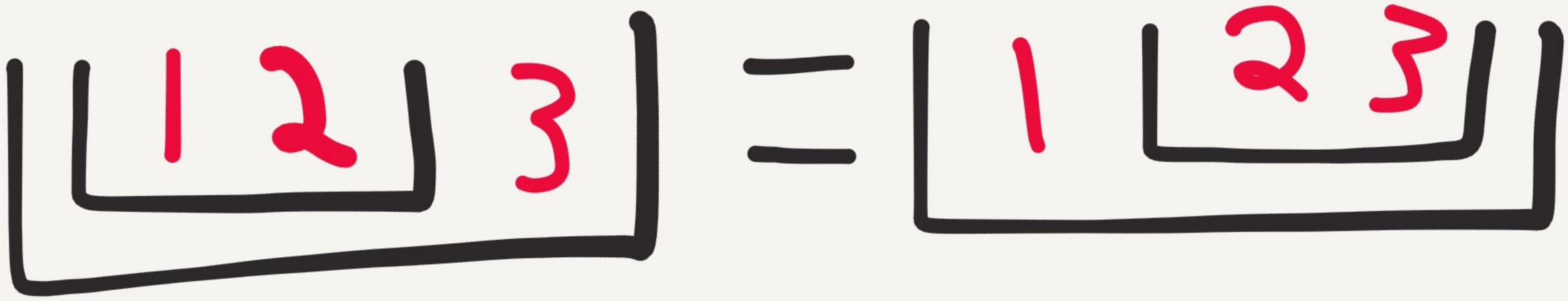
Reducers/combiners must be pure too!

But! Accumulators may be mutable

Hit me with
your best shot!

```
(fn [acc x]
  (.add acc x))
```

Reducers/Combiners must be associative



Reducers/combiners must be commutative

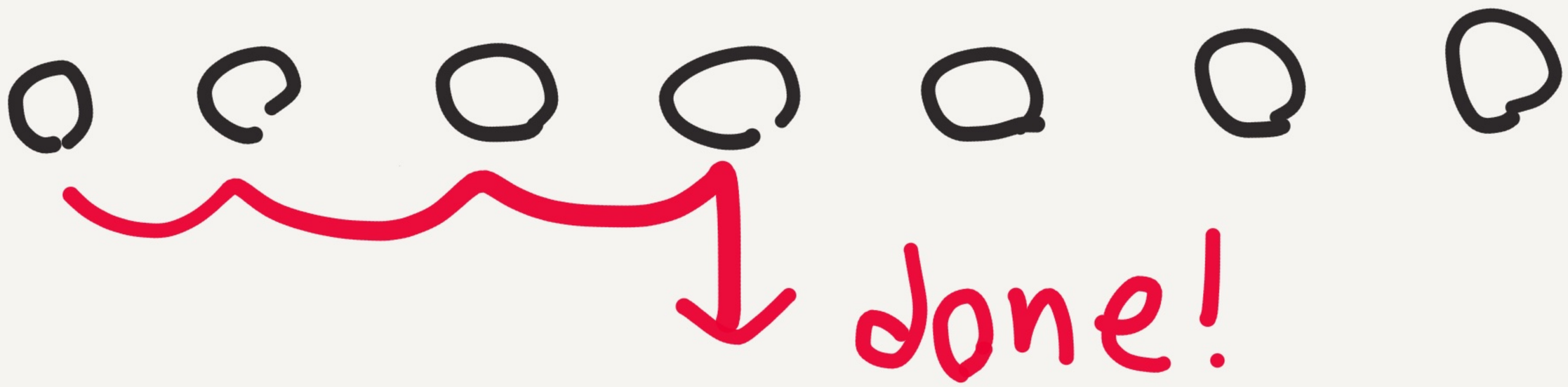
FREAKY FNOAY!



Algebird isn't commutative.

Should we order?

Like Transducers,
must short-circuit
(rednced x) valnes



API

```
(->> (t/map inc)
     (t/filter odd?)
     (t/take 100)
     (t/mapcat range)
     (t/into []))
```

BUT WAIT!

THERE'S MORE!



any

remove

Count

every?

frequencies

some

group-by

set

empty?

min

Keep

max

range

But that's NOT ALL!

MATH

Variance Covariance

mean stddev sum

quantiles correlation

CORRELATION MATRICES!

ALL for the
Low Low Cost
- of -
ONE PASS

Call now.

and get N

reductions for the

Cost of one!

```
(t/fuse
  {:ages (->> (t/map :age)
              (t/range))
   :cities (->> (t/map :city)
                (t/frequencies))})
```

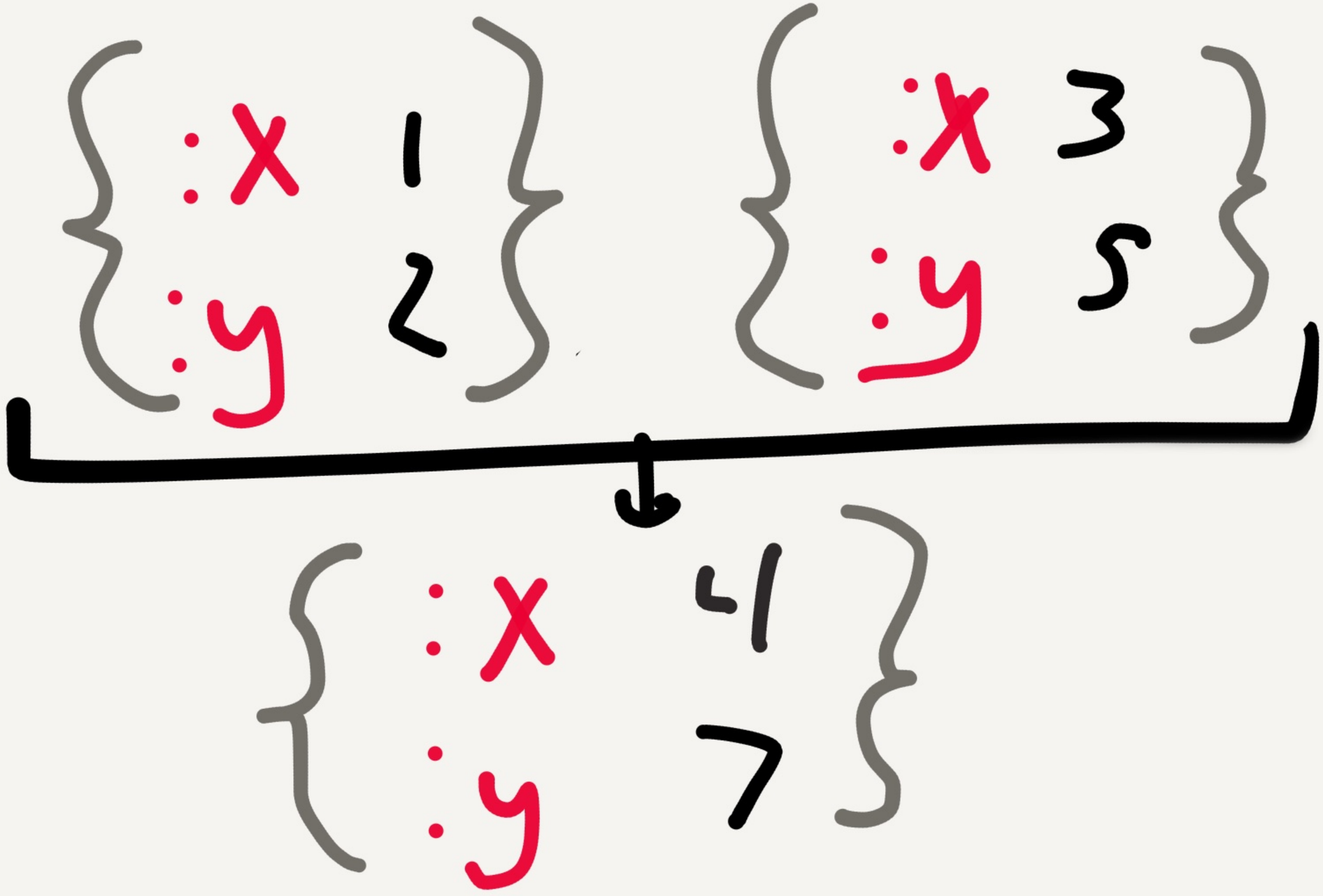
Yields...

```
{:ages [18, 105]
 :cities {"SF" 1158,
          "London" 957}}
```

```
(->> (t/map parse-str)
     (t/map :demo)
     (t/fuse {:ages ...
              :cities ...}))
```

Only need to parse once!

```
(->> (t/facet) (t/sum))
```





Recap

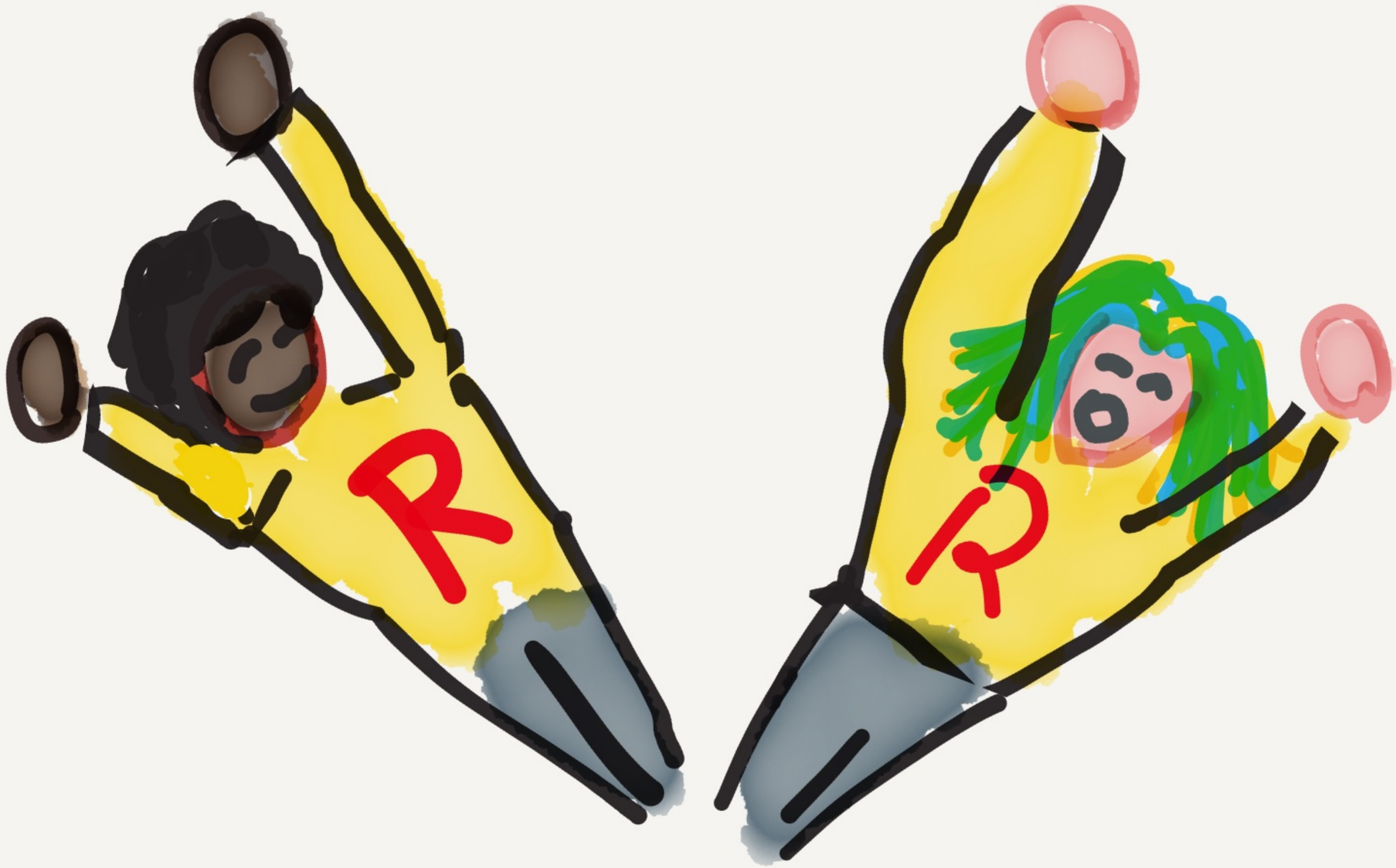
Stream Fusion



Collection Independent



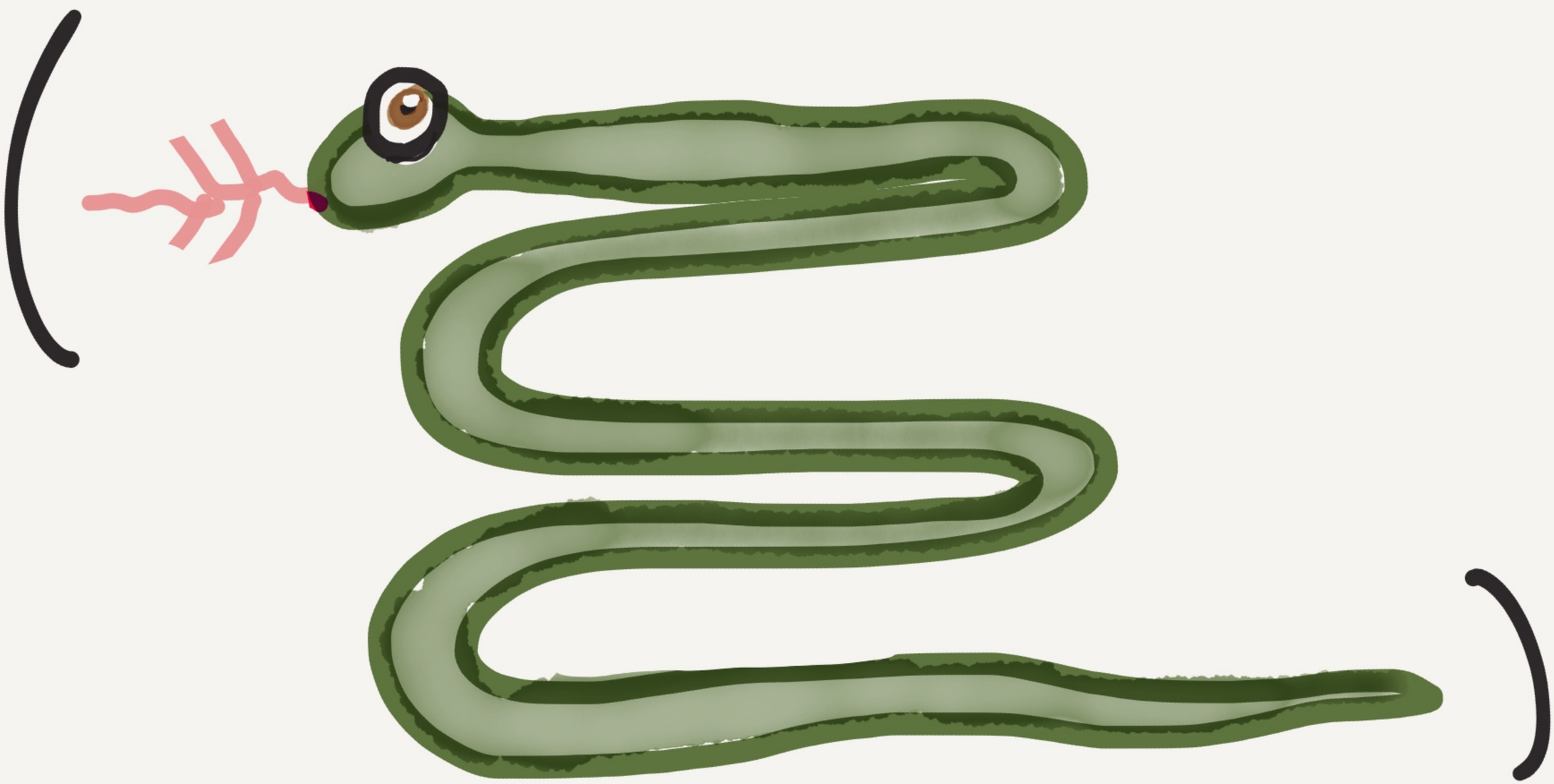
Parallel



Commutative



Seq-like API



THANKS!

- Natasha Whitney
- Zach Tellman
- Factual
- Tyler Schnett
- Brent Gannetta



github.com/aphyr/tesser

by

Kyle Kingsbury
Natasha Whitney