LECTURE DATAFLOW NETWORKS

Idea: how do work with infinite data

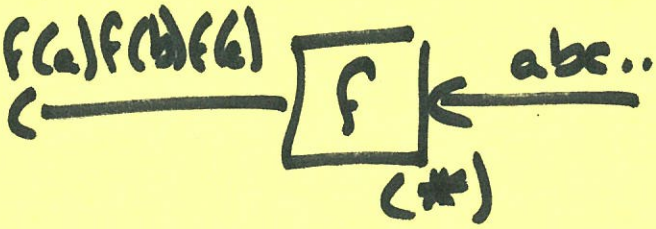
↳ streaming data/infinite sequences.

We build networks of computational nodes

Components

csd

map f *

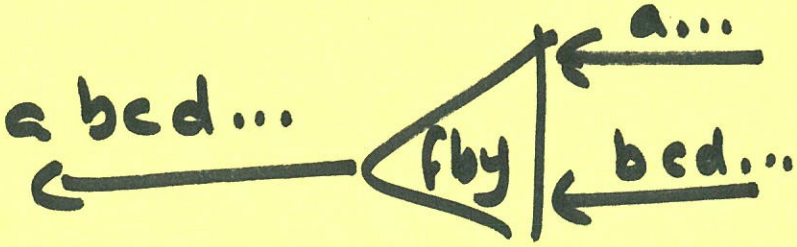


blocks until there's an input

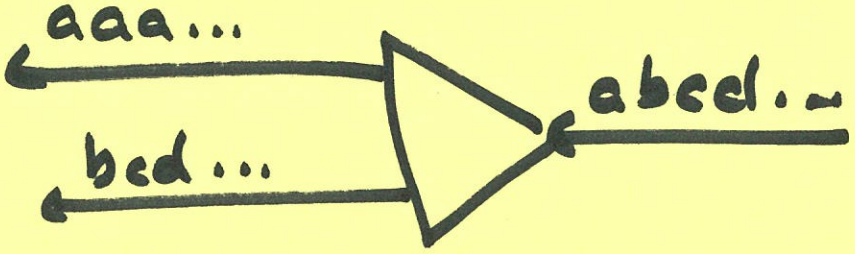
What can f do? Computable

~~Work of the Turing machine~~!

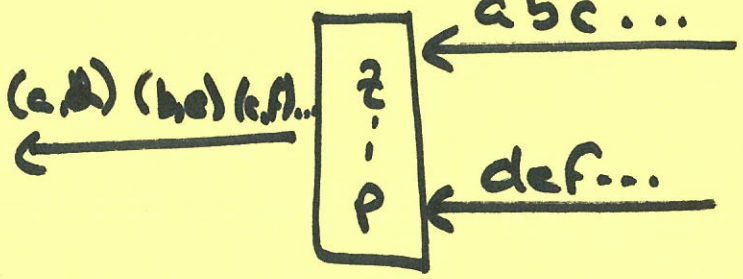
fby



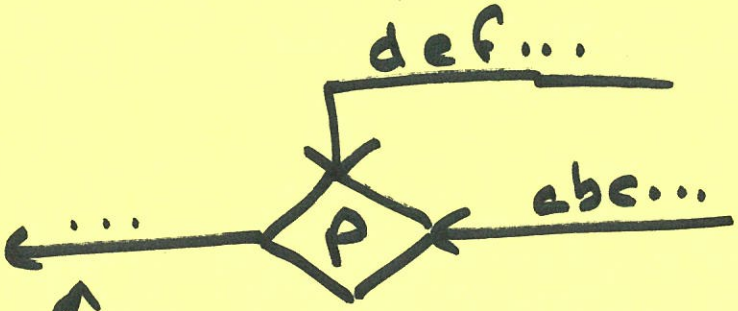
split



zip*



filter p

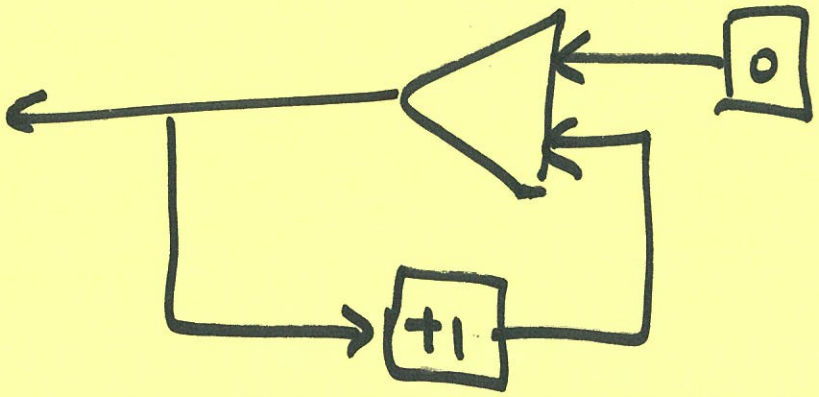


a if p a d is true
b if p b e is true
c if p c f is true

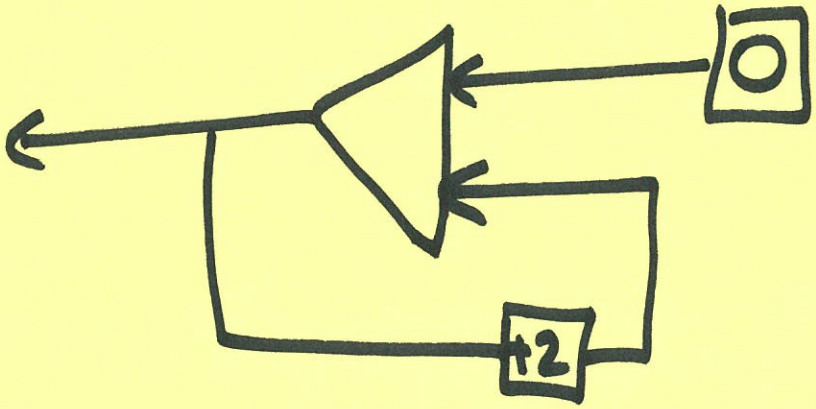
Could have more, but this is sufficient for our needs.

Examples

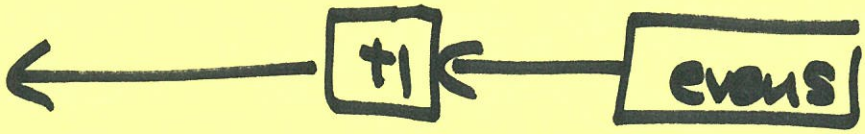
nats



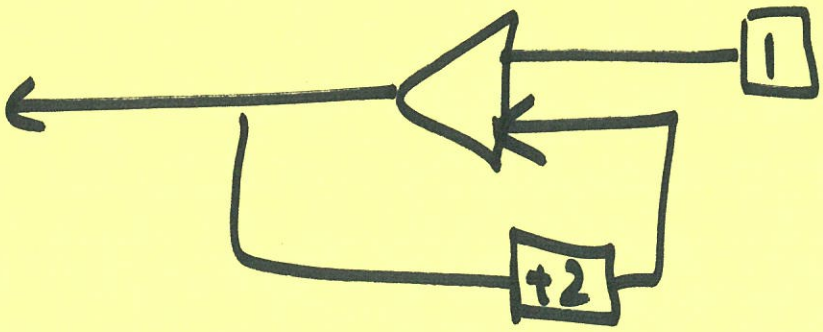
evens



odds

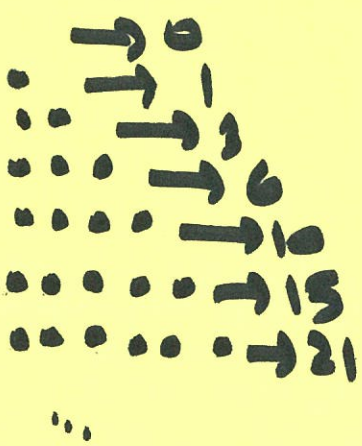


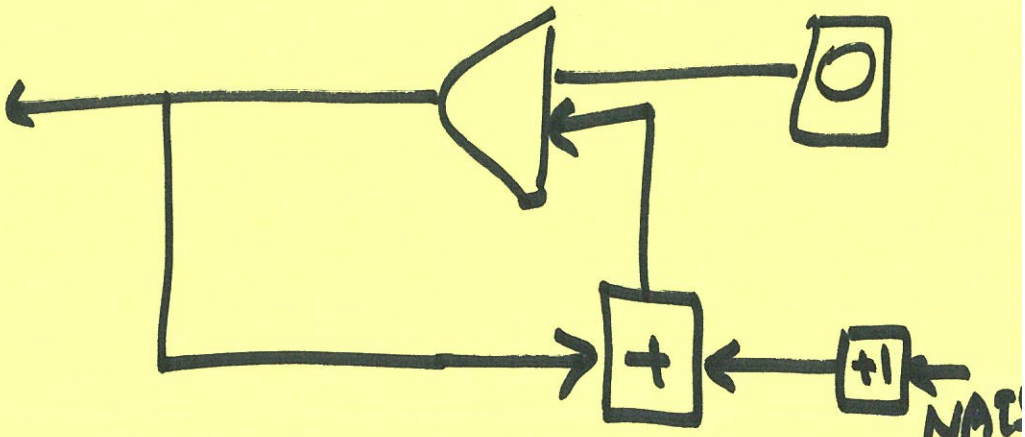
or



Indexing sequences

Δ #s:





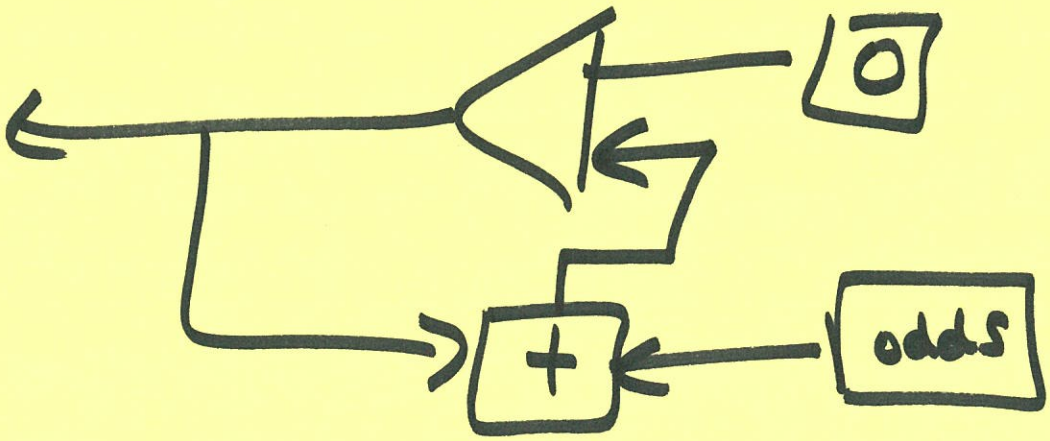
+ 0 1 3 6
+1 +2 +3 +4 +

OR:

	0	1	3	6	10	15	21	...
+	1	2	3	4	5	6	7	...
0 →	1	3	6	10	15	21	28	...

□ #s

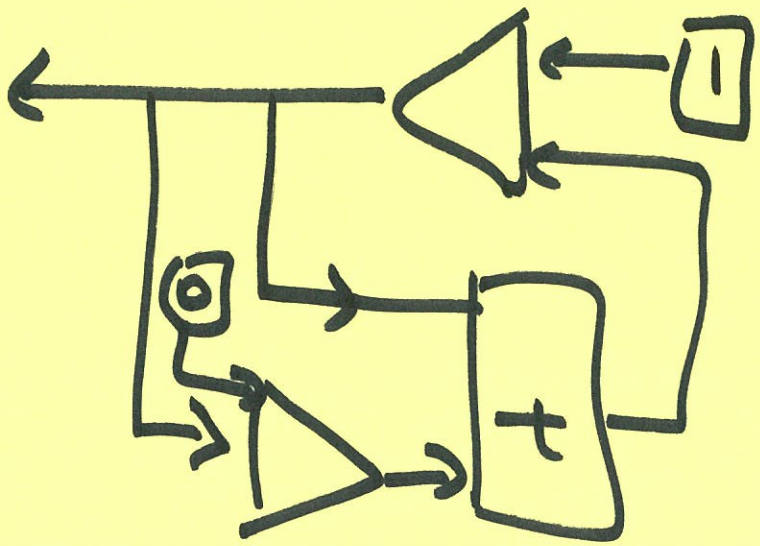
+ 0 1 4 9 16
+1 +3 +5 +7 +...



Fibonacci:

seed

1 1 2 3 5 8 13



1 1 2 3 5 8 13

+ 0 1 1 2 3 5 8

---

1 → 1 2 3 5 8 13 21

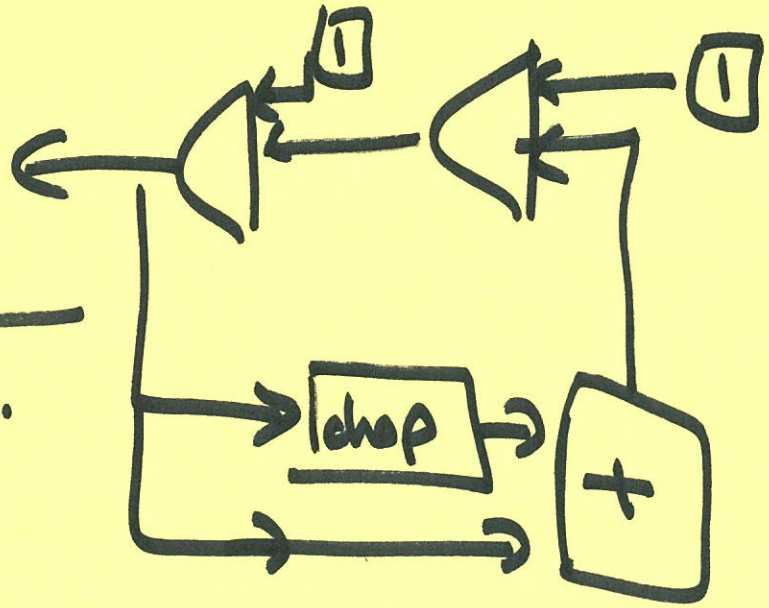
OR:

1 1 2 3 5 8 13

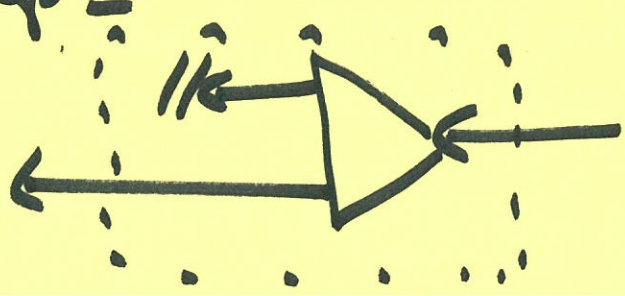
drop 1 2 3 5 8 13 21

---

1 1 2 3 5 8 13 21 ...



where drop ≡

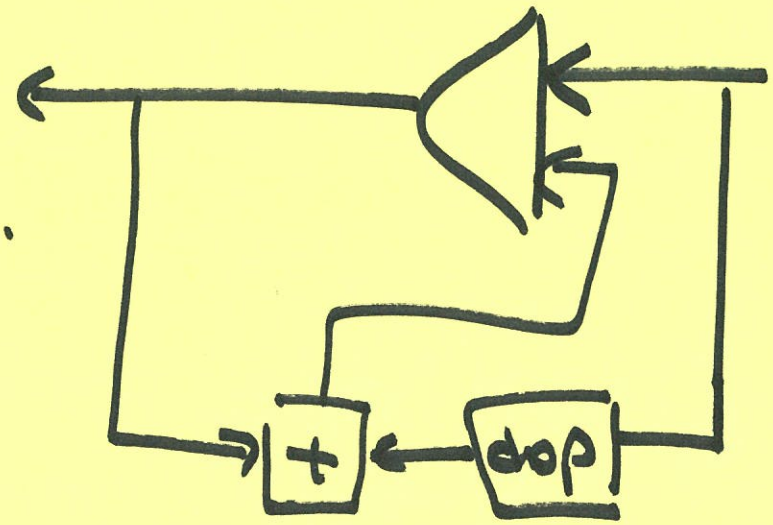


Partial sums

a b c d ...

⇓

a a+b a+b+c a+b+c+d ...



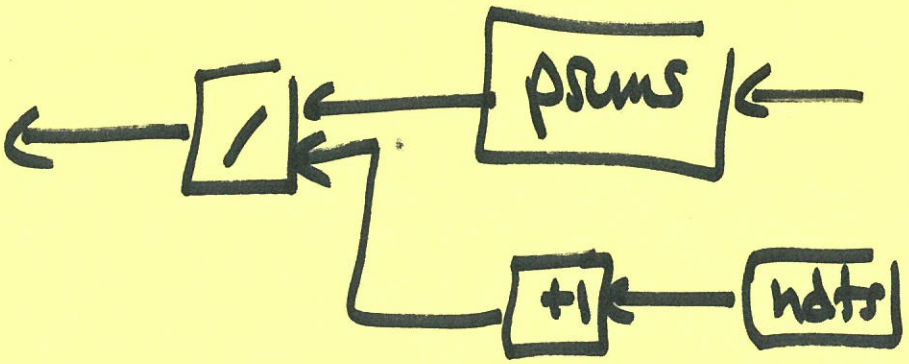
a a+b a+b+c a+b+c+d

+ b c d e

---

a+b a+b+c a+b+c+d ...

Running average



Sieve of ~~Eratosthenes~~ Eratosthenes

Sieve: remove all multiples from a stream.