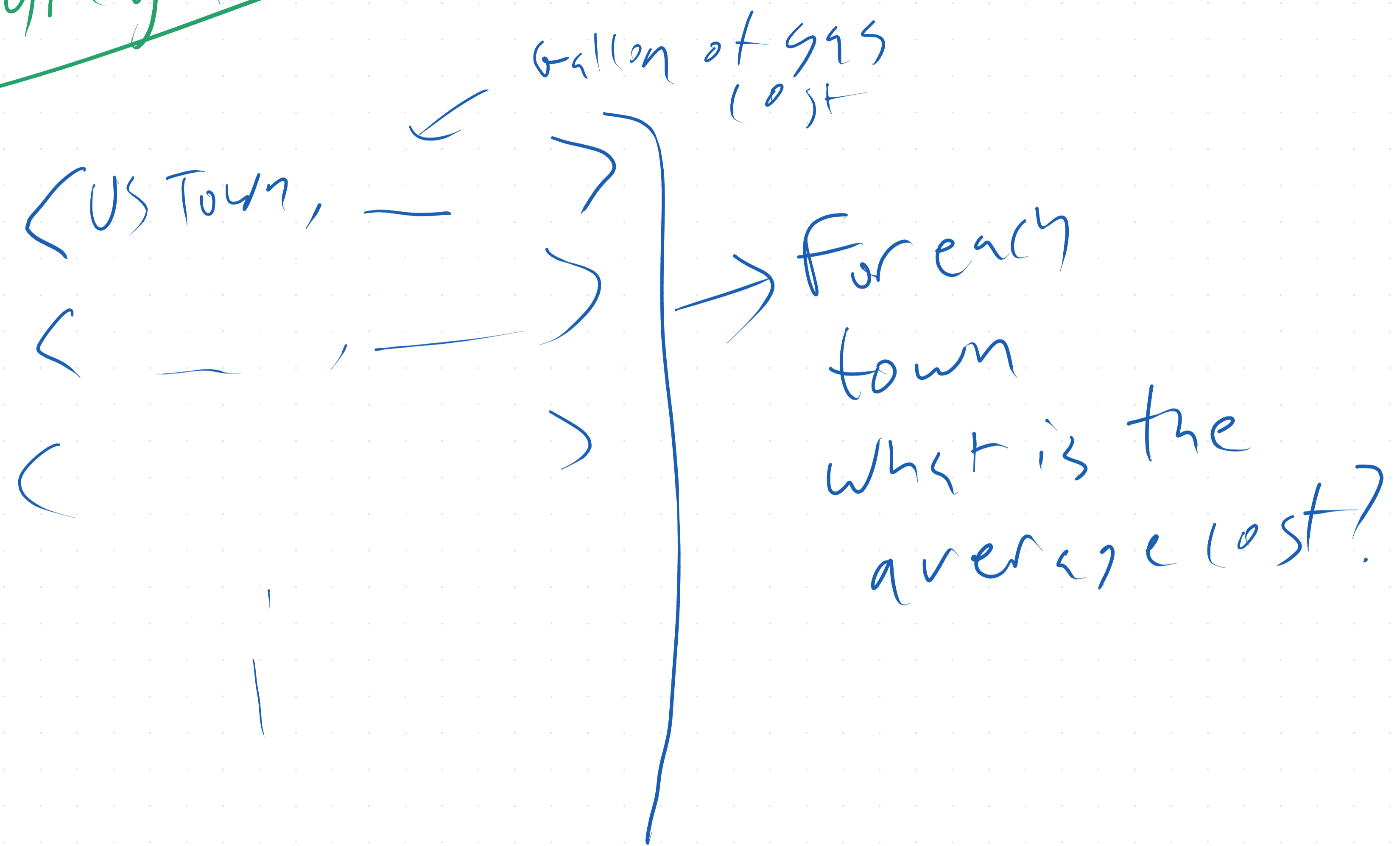
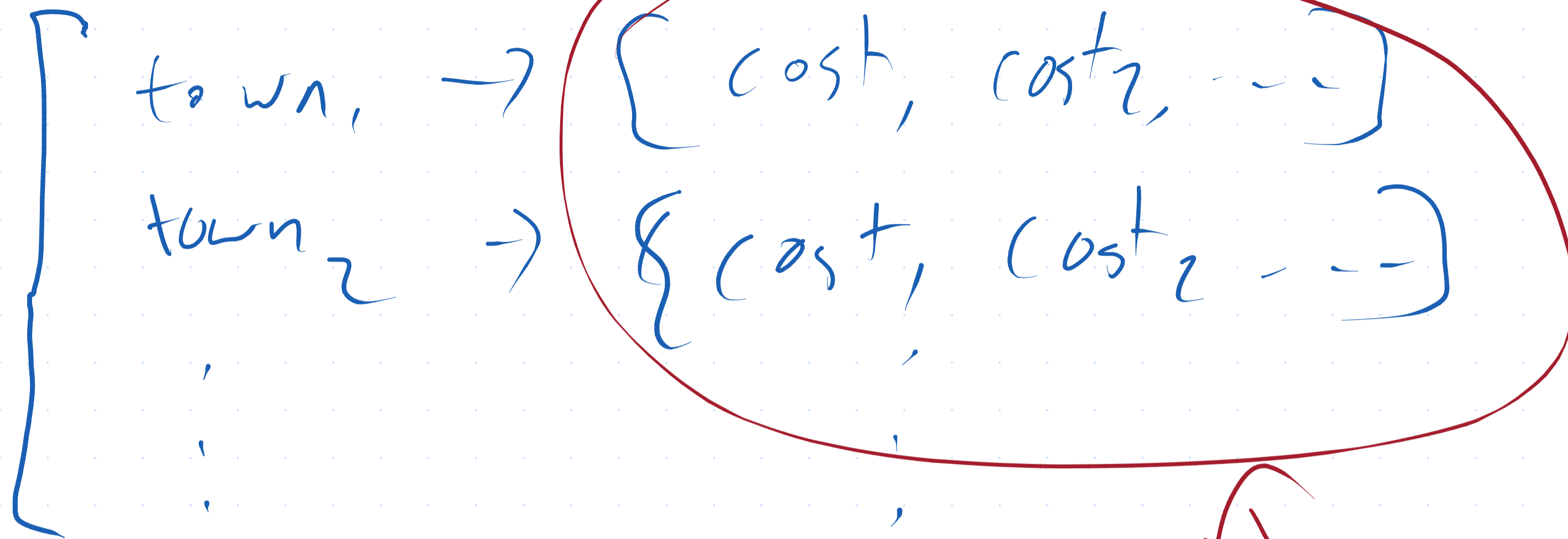


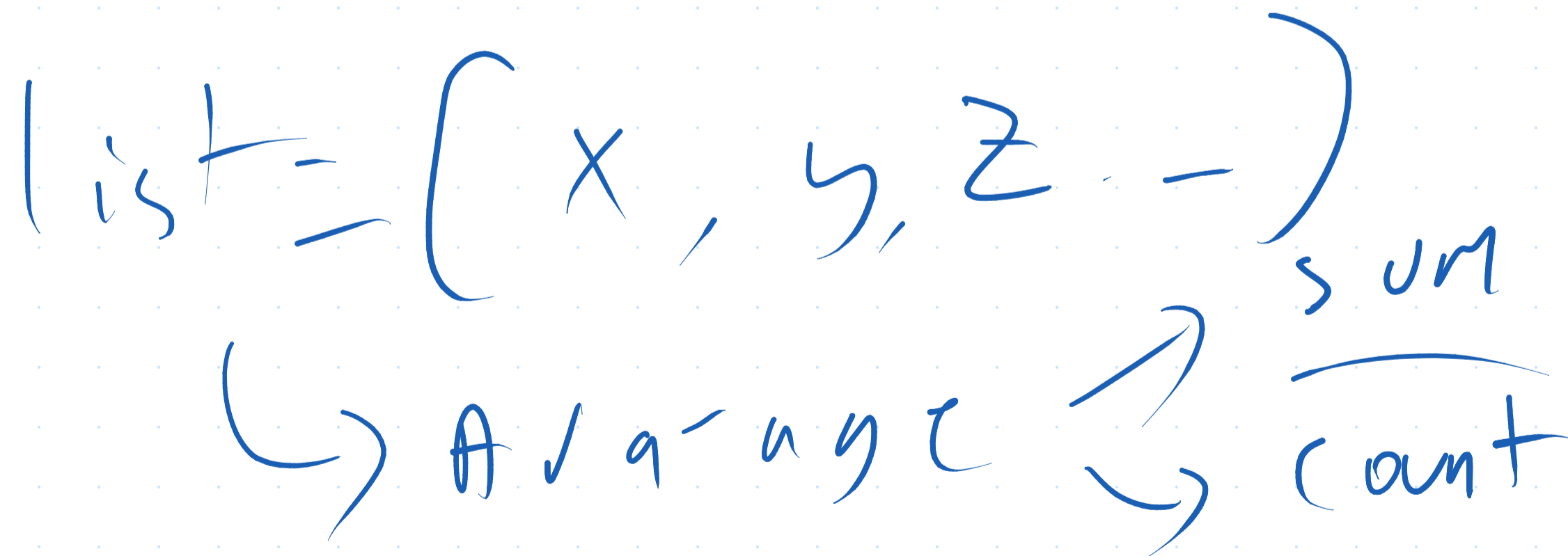
Aggregation

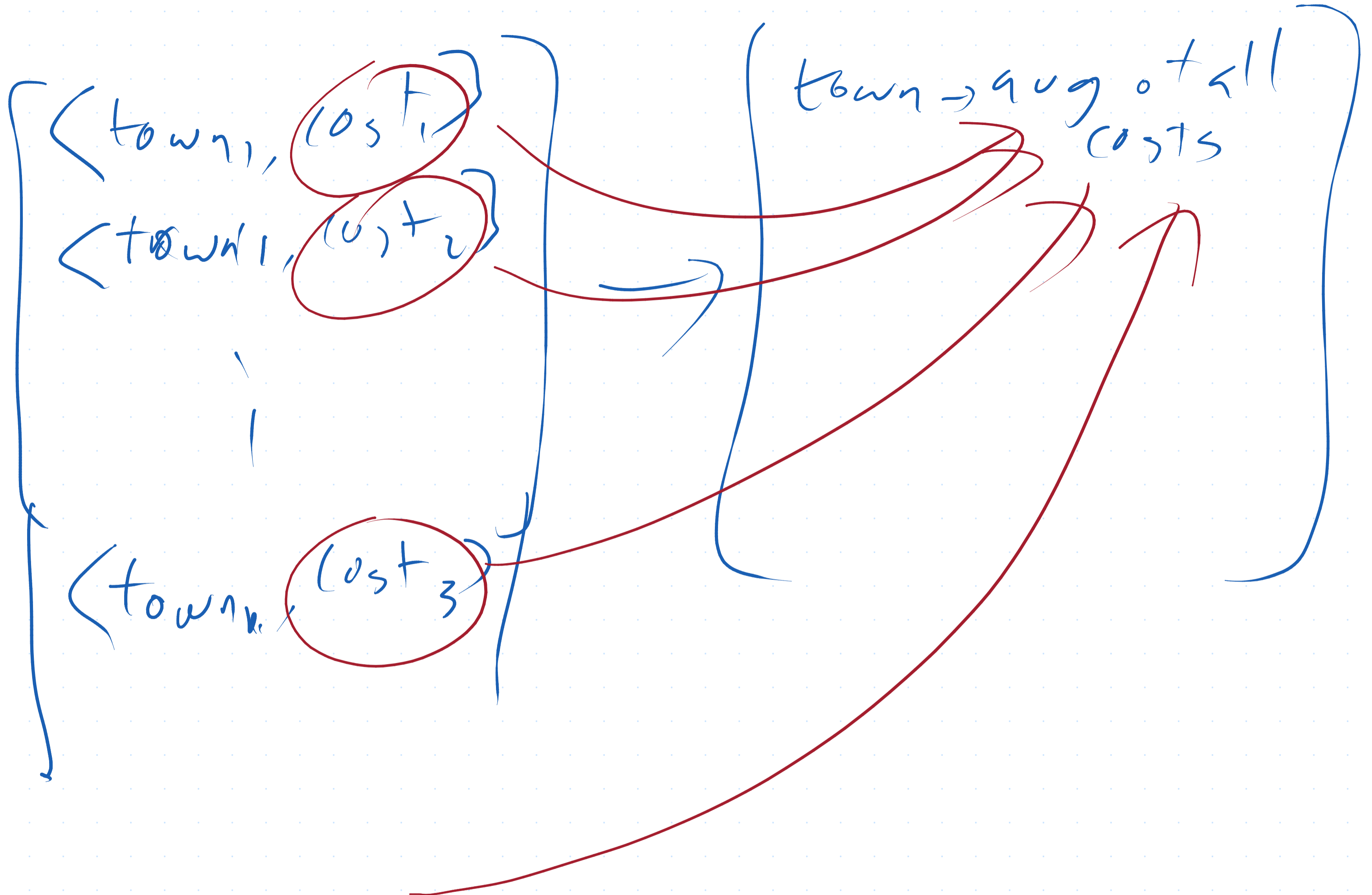


Hash
Table



Sum & Count





1. Scan through list

↳ find town in hash table (or add entry)

↳ add cost to sum

↳ add 1 to count

Memory

$O(\# \text{ of groups})$

(towns)



2. Scan through hash table

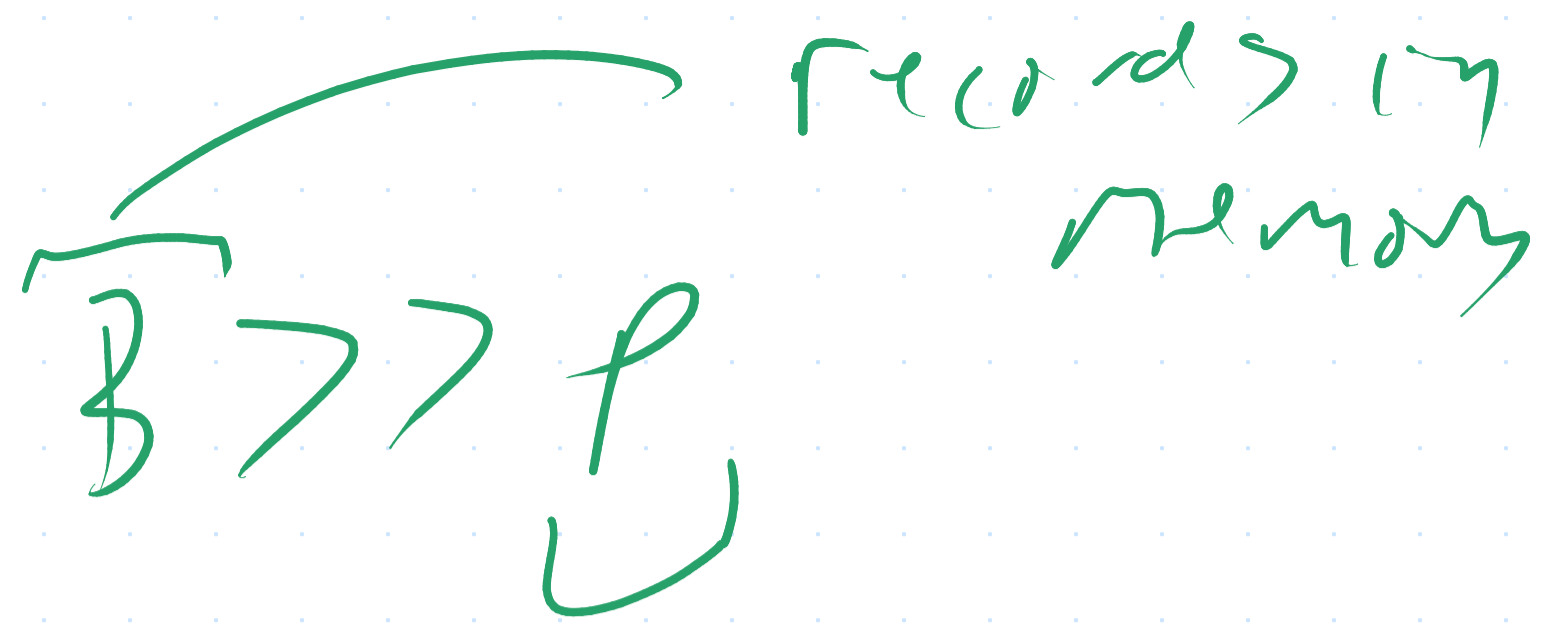
↳ compute average

Idea 1

Bit tree instead of hash table

$\log_p b$ per record
 $N \log_p b$ I/Os

Idea 2



$O(N \log_B N)$
I/Os for merge sort

one page
of records

Sort on town $\rightarrow O(B)$ memory

town, town, town \rightarrow sum, count $\rightarrow O(P)$ memory

town,
town,
town,

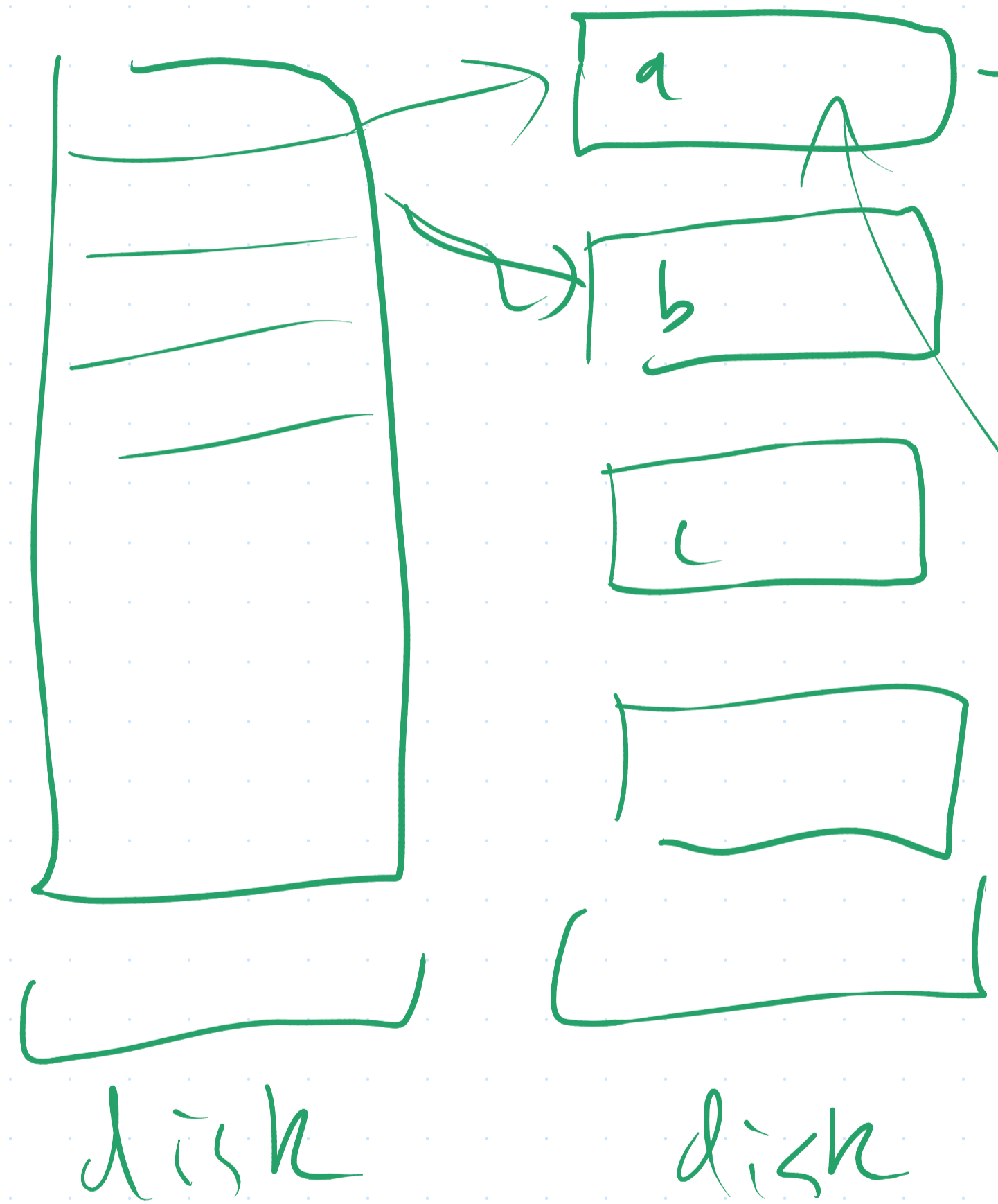
town₂

at this point
no more town, so write
out town, average &
free memory

Idea 3

① partition data (eg. on first letter)

shuffle



② do in memory algorithm w/ hash table

$O(\frac{m}{M})$ expected records per chunk

$O(NM)$ total I/Os

③ Read input file; create M output files

append to the "right" file

$O(N)$
memory
 $O(N)$ I/Os

same first letter

$O(N)$ I/Os total

$O(N) + O\left(\frac{6}{M}\right)$ memory

↑ expected

small edit
use $h(A) \% M$
to pick a chunk