MapReduce: Exercises

Exercise (See ex. 2.3.1.(b) of [LRU14])

Let S be a set of N integers represented by pairs (i, x_i), with $0 \leq i < N$, where i is the key of the pair and x_i is an arbitrary integer.

1. Design a 2-round MR algorithm to compute the arithmetic mean of the integers in S, using $O\left(\sqrt{N}\right)$ local space and $O(N)$ aggregate space. Carefully specify the input and output pairs of each round. You may assume that N is a global variable known to the algorithm.
2. Show how to modify the algorithm of the previous point to reduce the local space bound to $O\left(N^{1/4}\right)$, at the expense of an increased number of rounds.
3. Can you generalize the previous point to attain $O(N^{\epsilon})$, for any $\epsilon \in (0, 1/2)$?

① INPUT : $\{(i, x_i) : 0 \le i < N,\ x_i \text{ integer}\}$

OUTPUT : $(0, \text{mean}(S) = (1/N) \cdot \sum_{i=0}^{N-1} x_i)$

Round 1

Map Phase : $\forall 0 \le i < N \quad (i, x_i) \to (i \bmod \sqrt{N}, x_i)$

Reduce Phase: for each $0 \le j < \sqrt{N}$ separately, let

L_j = list of integers x from intermediate pairs (j, x)

$(j, L_j) \to (0, \text{sum}_j = \sum_{x \in L_j} x)$

Round 2

Map Phase : empty

Reduce Phase: let $L_0 = sum_0, sum_1, \ldots, sum_{\sqrt{N}-1}$ be the list of pairs produced by R1

$$(0, L_0) \longrightarrow \left(0, \underbrace{mean(S) = \frac{1}{N} \sum_{sum_j \in L_0} sum_j}_{\text{Global variable}}\right)$$

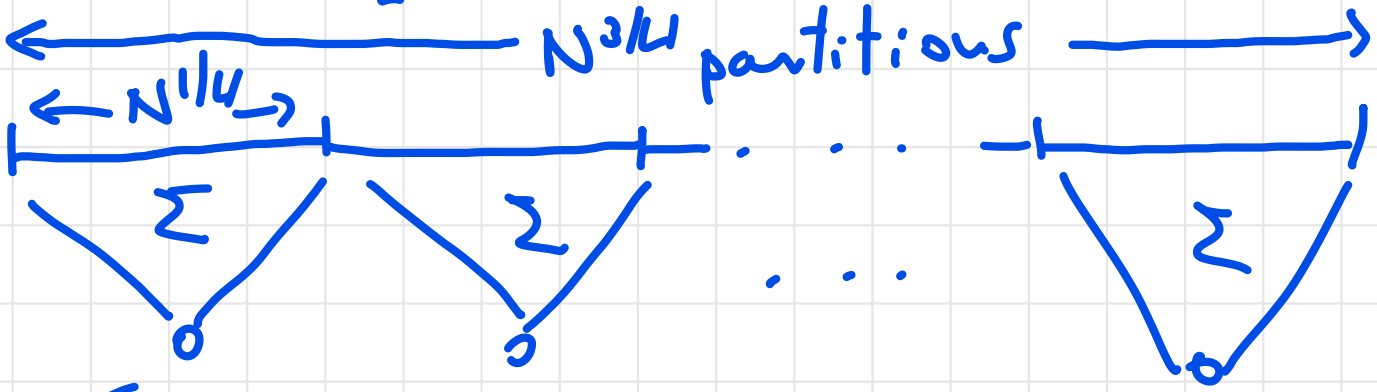
obs. If we want to avoid possible overflows due to very large numbers, we can compute the mean locally in each partition in R1, together with the size of the partition, and then, in R2, compute the "weighted" mean of the $\sqrt{N}$ local means

$M_L = O(\sqrt{N})$ since in the reduce phases of the 2 rounds, each reducer adds up to $\sqrt{N}$ integers

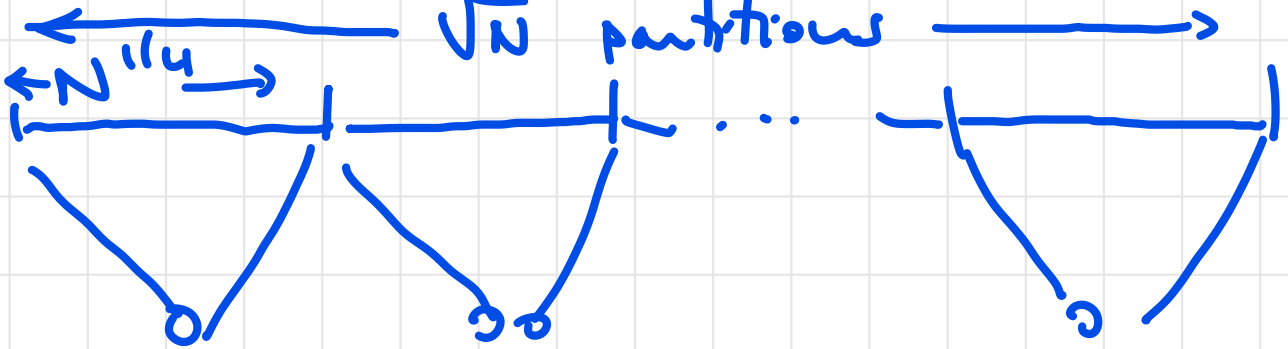
$M_A = O(N)$ since the number of input, output, and intermediate pairs in each round is $O(N)$

(2) OBJECTIVE : $M_L = O(N^{1/4})$

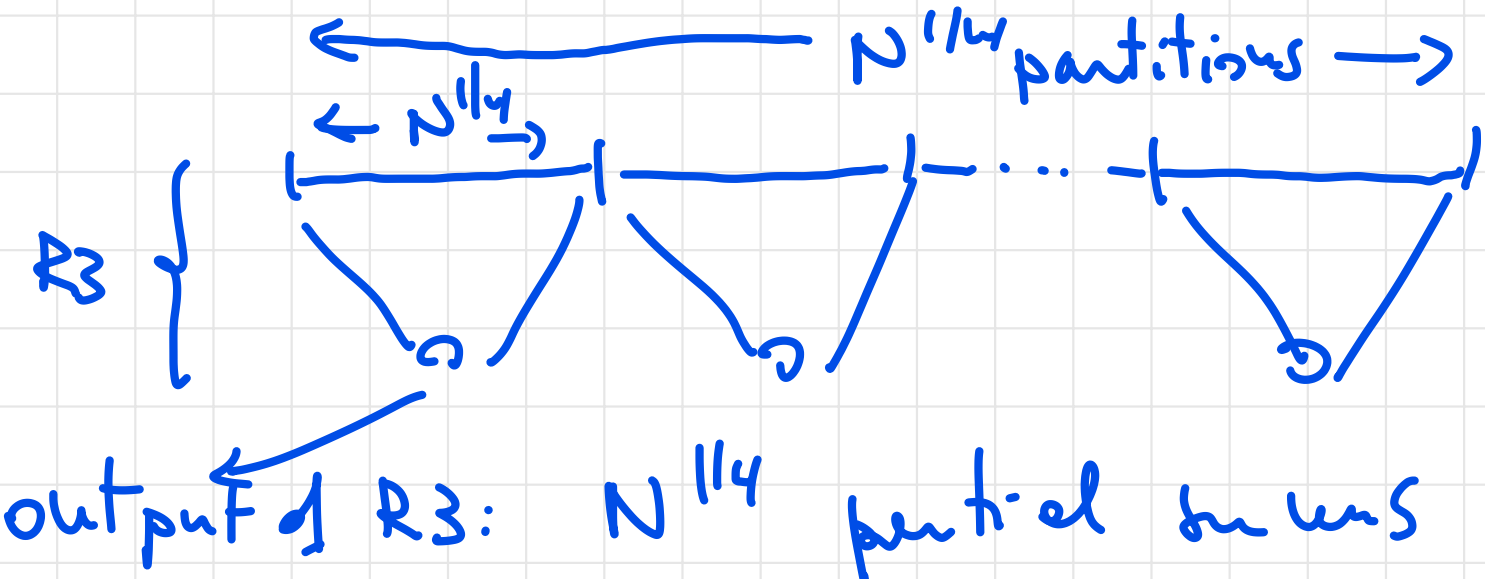
IDEA



Output of R1 : $N^{3/4}$ partial sums



Output of R2 : $N^{1/2}$ partial sums



output of R3: $N^{1/4}$ partial sums

R4 : Compute mean (S) by aggregating the $N^{1/4}$ partial sums produced by R3

Round 1

Map Phase: $\forall 0 \le i < N : (i, x_i) \rightarrow (i \bmod N^{3/4}, x_i)$

Reduce Phase: $\forall 0 \le j < N^{3/4}$ separately let

L_j = list of integers x from intermediate pairs (j, x)

$(j, L_j) \rightarrow \left(j, \sum_{x \in L_j} x\right)$

Round 2

Map Phase : $\forall\ 0 \le j < N^{3/4}$ $(j, s) \to (j \bmod N^{1/2}, s)$

Reduce Phase : $\forall\ 0 \le j < N^{1/2}$ separately, let

L_j = list of integers s in intermediate pairs (j, s)

$(j, L_j) \to \left(j, \sum_{s \in L_j} s\right)$

Round 3
Round 4
} Exercise

Analysis

$R = 4$

$M_L =$ for each round

* Map Phase: $O(1)$
* Reduce Phase: $O(N^{1/4})$

$\Rightarrow M_L = O(N^{1/4})$

$M_A = O(N)$ since the number of input, output and intermediate pairs in each round is $O(N)$

③ $N \to N^{1-\varepsilon} \to N^{1-2\varepsilon} \cdots N^{1-i\varepsilon} = N^{\varepsilon}$ $\quad i+1$ round $i+1 = \frac{1}{\varepsilon}$

Exercise (See ex. 2.3.1.(d) of [LRU14])

Let S be a set of N integers represented by pairs (i, x_i), with $0 \leq i < N$, where i is the key of the pair and x_i is an arbitrary integer. We want to design an efficient MR algorithm to compute the number D of distinct integers in S.

1. Design a simple 2-round MR algorithm to compute D and analyze its local and aggregate space requirements. Under what circumstances is the local space proportional to N, thus not meeting the design goals of MR algorithms?
2. Design a better MR algorithm to compute D in $O(1)$ rounds, using $O\left(\sqrt{N}\right)$ local space and $O(N)$ aggregate space.

① INPUT: $\{(i, x_i) : 0 \le i < N, x_i \text{ integer}\}$

OUTPUT $(0, \text{count})$ count = # distinct integers x_i's

IDEA

R1: eliminate duplicates

R2: Count how many integers (all distinct) are left

Round 1

Map Phase $\forall\ 0 \le i < N \quad (i, x_i) \to (x_i, i)$

Reduce Phase: $\forall$ integer x seperately let

L_x = list of i's from intermediate pairs (x_i, i) with $x_i = x$

$(x, Lx) \rightarrow (0, x)$

Round 2

Map these: empty

Reduce these: let L_0 the list of integers from pairs $(0, x)$ produced by R1

$(0, L_0) \rightarrow (0, |L_0|)$ → # distinct integers

Analysis

$M_A = O(N)$ since the number of input, output and intermediate pairs in each round is $O(N)$

$M_L =$

* N_1 = max # of copies for integer
* N_2 = # distinct integers

$M_L = O(\max(N_1, N_2)) = O(N_1 + N_2)$

and both N_1 and N_2 can be proportional to N in the worst case.

② OBJECTIVE $M_L = O(\sqrt{N})$

IDEA: * Duplicate removals : 2 rounds
* Count distinct : 2 rounds

Round 1

Map Phase: $\forall\, 0 \le i < N \quad (i, x_i) \to (i \bmod \sqrt{N}, x_i)$

Reduce Phase: $\forall\, 0 \le j < \sqrt{N}$ separately, let

L_j = list of integers x from intermediate pairs (j, x)

$(j, L_j) \to \{(j, x) : \forall$ distinct integer $x \in L_j\}$

AT THIS POINT: there are $\le \sqrt{N}$ copies of each integer

Round 2

Map Phase: ∀ pair (j, x) from R1 $(j, x) \to (x, j)$

Reduce Phase: ∀ integer x, let

L_x = list of indices j of intermediate pairs (x, j)

$(x, L_x) \to (x, j)$ with j an arbitrary index from L_x

AT THIS POINT

* Only 1 copy of each integer is left
* $\leq \sqrt{N}$ pairs with index j

Round 3

Map phase $\forall$ pair (x, j) : $(x, j) \rightarrow (j, x)$

Reduce phase $\forall\ 0 \leq j < \sqrt{N}$ let

L_j : list of integers from intermediate pairs (j, x)

$(j, L_j) \rightarrow (0, c_j = |L_j|)$

Round 4

Map Phase: empty

Reduce Phase: let $L_0 = c_0 c_1 \cdots c_{\sqrt{N}-1}$

$$(0, L_0) \rightarrow \left(0, \sum_{c_j \in L_0} c_j\right)$$

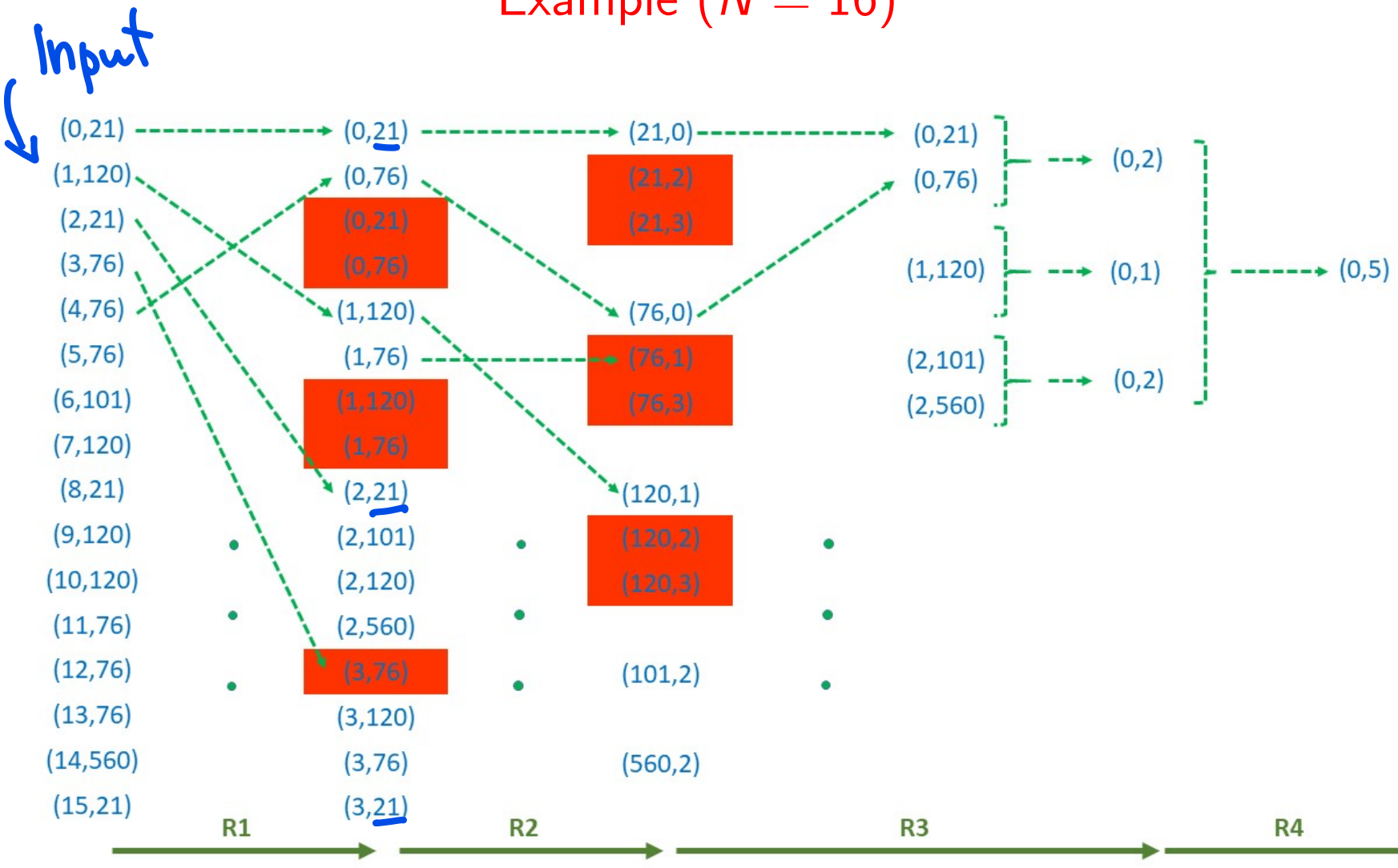
→ # distinct integers in the input set

Analysis

$M_A = O(N)$ usual argument

$M_L = O(\sqrt{N})$ because of the initial partition and of the fact that after $R1 \leq \sqrt{N}$ copies of each integer were left

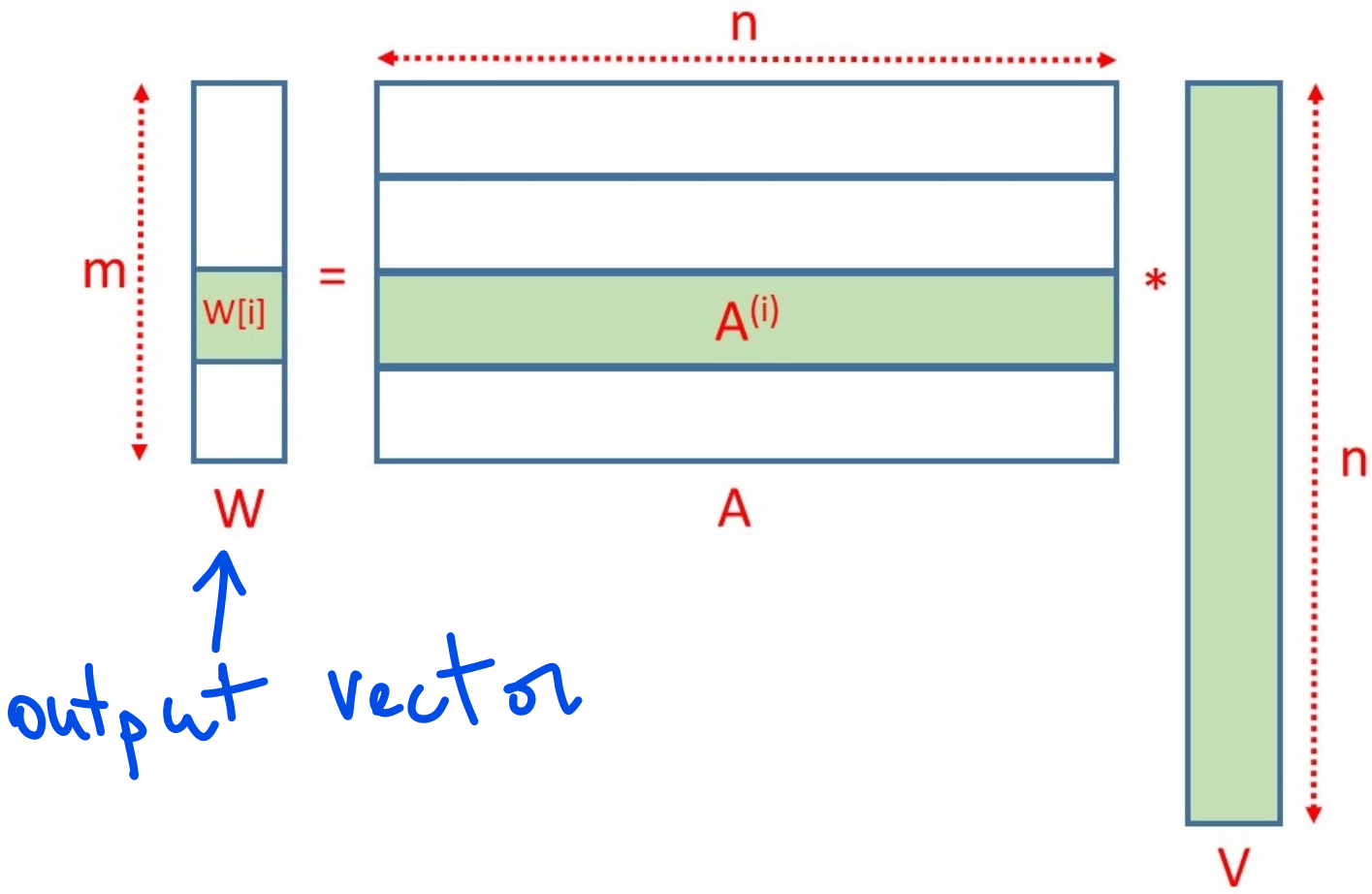
Example ($N = 16$)


Input
(0,21)
(1,120)
(2,21)
(3,76)
(4,76)
(5,76)
(6,101)
(7,120)
(8,21)
(9,120)
(10,120)
(11,76)
(12,76)
(13,76)
(14,560)
(15,21)
(0,21)
(0,76)
(0,21)
(0,76)
(1,120)
(1,76)
(1,120)
(1,76)
(2,21)
(2,101)
(2,120)
(2,560)
(3,76)
(3,120)
(3,76)
(3,21)
(21,0)
(21,2)
(21,3)
(76,0)
(76,1)
(76,3)
(120,1)
(120,2)
(120,3)
(101,2)
(560,2)
(0,21)
(0,76)
(1,120)
(2,101)
(2,560)
(0,2)
(0,1)
(0,2)
(0,5)
R1
R2
R3
R4

Exercise

Design an $O(1)$-round MR algorithm for computing a matrix-vector product $W = A \cdot V$, where A is an $m \times n$ matrix, V is an n-vector, and $m \leq \sqrt{n}$. Your algorithm must use $o(n)$ local space and linear (i.e., $O(mn)$) aggregate space.

How would your solution change if m were larger?



INPUT $A = \{((i,j), A[i,j]) : 0 \le i < m \quad 0 \le j < n\}$

$V = \{((j,-1), V[j]) : 0 \le j < n\}$

OUTPUT $W = \{(i, W[i]) : 0 \le i < m\}$

1-round algorithm (space inefficient)

2-round algorithm (space efficient)

$W[i] = A^{(i)} \circ V$ → inner product

1-round algorithm

Map phase

$((i,j), A[i,j]) \rightarrow (i, ((i,j), A[i,j]))$

$((j,-1), V[j]) \rightarrow \{(i, ((j,-1), V[j])) : 0 \le i < m\}$

$\Rightarrow$ m copies of V

Notation: $V^{(i)} \equiv$ copy i of V $0 \le i < m$

$W[i] = A^{(i)} \circ V^{(i)}$

$A^{(i)}$: i-th row of A

$V^{(i)}$: i-th copy of V

Reduce Phase $\forall\ 0 \le i < m$ separately, let

L_i = list of values from intermediate pairs with key $i \Rightarrow L_i$ contains the representation of $A^{(i)}$ and $V^{(i)}$

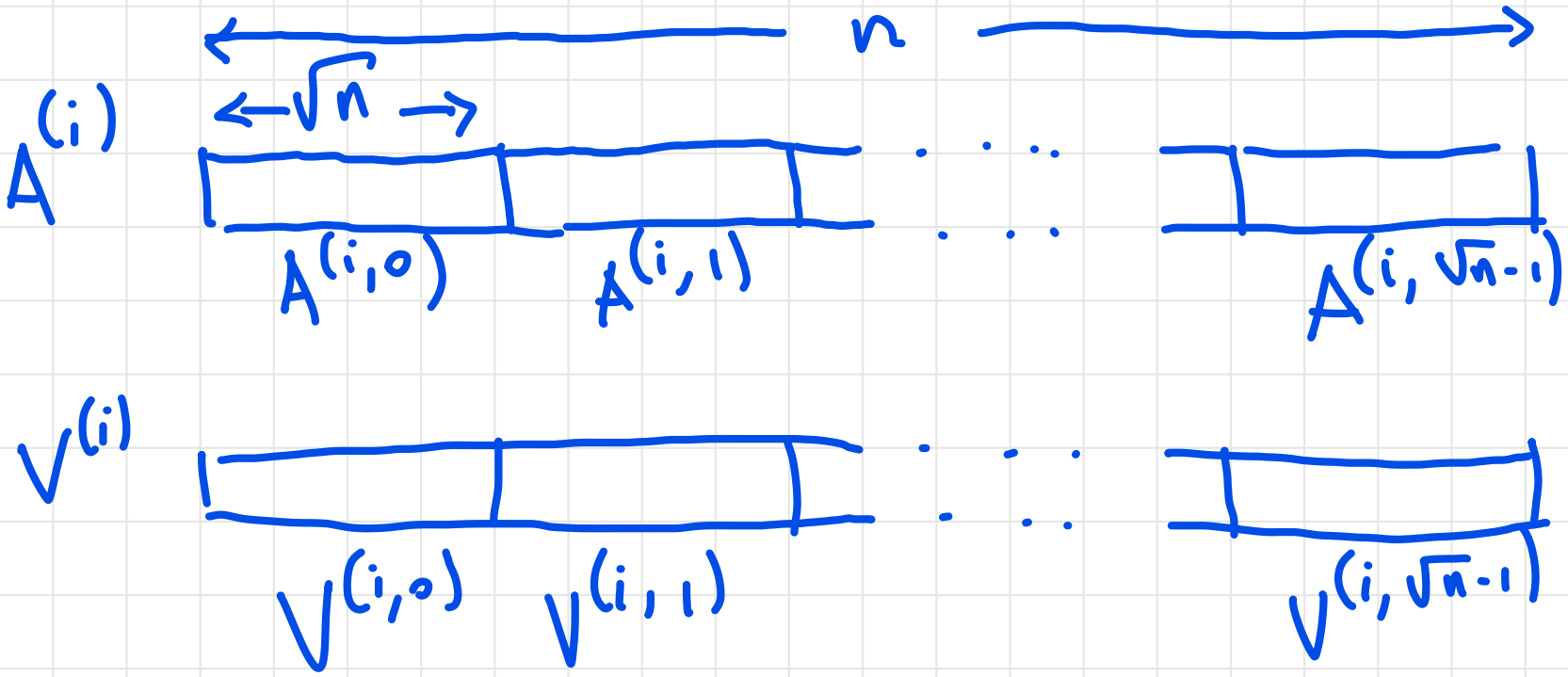
$$(i, L_j) \rightarrow (i, W[i] = A^{(i)} \circ V^{(i)})$$

$$M_A = O(n \cdot m)$$

$$M_L = O(m + n) = O(n)$$

Map Phase, Reduce Phase

OBJECTIVE : $M_L = o(n)$ in particular $M_L = O(\sqrt{n})$



$$\Rightarrow |A^{(i,s)}| = |V^{(i,s)}| = \sqrt{n} \quad \forall\ 0 \le s < \sqrt{n}$$

Define $W_s[i] = A^{(i,s)} \circ V^{(i,s)}$

$$\Rightarrow W[i] = \sum_{s=0}^{\sqrt{n}-1} W_s[i]$$

Round 1

Map Phase

* Replicate each entry of V m times (as before)
* Assign key (i,s) to each entry of segment s of $A^{(i)}$ (i.e., $A^{(i,s)}$) and to each entry of segment s of the i-th copy of V (i.e., $V^{(i,s)}$)

Reduce Phase $\forall\, 0 \le i < m$ and $\forall\, 0 \le s < \sqrt{n}$

Compute $W_s[i] = A^{(i,s)} \circ V^{(i,s)}$

returning $(i, W_s[i])$

Round 2

Map phase : empty

Reduce phase:

Compute $\left(i, W[i] = \sum_{s=0}^{\sqrt{n}-1} W_s[i]\right)$

These are the values of the pairs produced by R1 with key i

Exercise: fill in the details

Analysis

$M_A = O(m \cdot n)$ as before

M_L :

* Map phase of R1 : $O(n)$ ← To create copies of V
* Reduce phase of R1 : $O(\sqrt{n})$ ← to compute the inner products of segments of length $\sqrt{n}$
* Map phase of R2 : —
* Reduce phase of R2 : $O(\sqrt{n})$

$\Rightarrow M_L = O(\max\{m, \sqrt{n}\}) = O(\sqrt{n})$ by the hypothesis $m \leq \sqrt{n}$