P1: mycellline: [0 1 2 3 4 5]

size: 2

myrecvdata: [- -]

vsize = 2

Allvec: [x0 x1 x2 x3]

recvdata size and recvdatsz fb — (1)

vsize = 2
vsize2 = 0.

[2 2 0] | [0 2 4]

runptr = [0 2 2 4 8]

ecount and edispls

[2 6 0] | [0 2 8]

allvect: [x0 x1 x2 x3].

P1: vsize = 2, size = 2.

myrecvdata: [x2 x3]

(myNum2) = 6.

myRowptr = [- - -] = [2 4 -]

myptroff = 2.

i = 0, 1

myrowptr[0] = 2 - 2 = 0. [0 4 -]

i = 1: myrowptr[1] = 4 - 2 = 2 [0 2 -]

myrowptr[2] = 6

[0 2 6] myRowPtr

myColInd:

sColInd: [0 2 2 3 0 1 2 3] → Scatter myColInd: [2 3 0 1 2 3]

1 0 1 0
0 0 0 0
0 0 3 3
4 4 4 4
(myColInd)

vals: [1 1 3 3 4 4 4 4] → Scatter (myVal+1): = [3 3 4 4 4 4]

prepareRemoteVec:

→ myColInd: [2 3 0 1 2 3]
myNnz = 6
nrows = 4
vsize = 2
myRecvNode: [x1 x3]

→ size = 2, lstart = 2, lend = 4.

tempRemoteCount: [- - -]

remoteVecCount = [0 0 0] → [1 0 0] → [2 0 0]
nRv = 1. nRv = 2.

remoteVecIndex: [- - - - - -]

numRemVec = 0.

i = 0 — 5

(i) i = 0: if((2 < 2) || (2 >= 4)) ✗

(ii) i = 1: if((3 < 2) || (3 >= 4)) ✗

(iii) i = 2: if((0 < 2) || (0 >= 4)) ✓

(iv) i = 3: if((1 < 2) || (1 >= 4)) ✓

(v) i = 4: if((2 < 2) || (2 >= 4)) ✗

(vi) i = 5: if((3 < 2) || (3 >= 4)) ✗

remove vec idx = [0 _ _] → [0 2 _] → [0 2 2]

temp rem vec len = [0 _ _] [0 0 _] [2 0 0]

myvecsorted = [x2 x3 _ _]

2+2 = 4

prec = 0

i = 0 — 5

(i) i = 0:
if((2 < 2) || (2 >= 4)) ✗

(ii) i = 1:
if((3 < 2) || (3 >= 4)) ✗

(iii) i = 2
if((0 < 2) || (0 >= 4)) ✓
prec = 0.
remvecindex[0 + 0] = 0. remvecindex[0] = 0.
myvec[2] = 2 + 0 + 0 = 2

myvec: [2 3 0 1 2 3]

↓

[0 3 0 1 2 3]

↓

[0 1 0 1 2 3]

↓

[0 1 2 1 2 3]

(iv) i = 3
if ((1 < 2) || (1 >= 4)) ✓ [0 1 2 3 2 3]
proc = 0
genvecindex[0 + 1] = 1 genvecindex[1] = 1.
myvecindex[3] = 2 + 0 + 1 = 3.

(v) i = 4
if ((2 < 2) || (2 >= 4)) ✗ [0 1 2 3 0 3]

(vi) i = 5
if ((3 < 2) || (3 >= 4)) ✗ [0 1 2 3 0 1]

New modified myvecindex = ([0 1 2 3 0 1])

nrv = 2

genvec count = [2 0 0]
genvec ptr = [0 2 2]
genvec index = [0 1 _ _ _ _]

Proc 0:

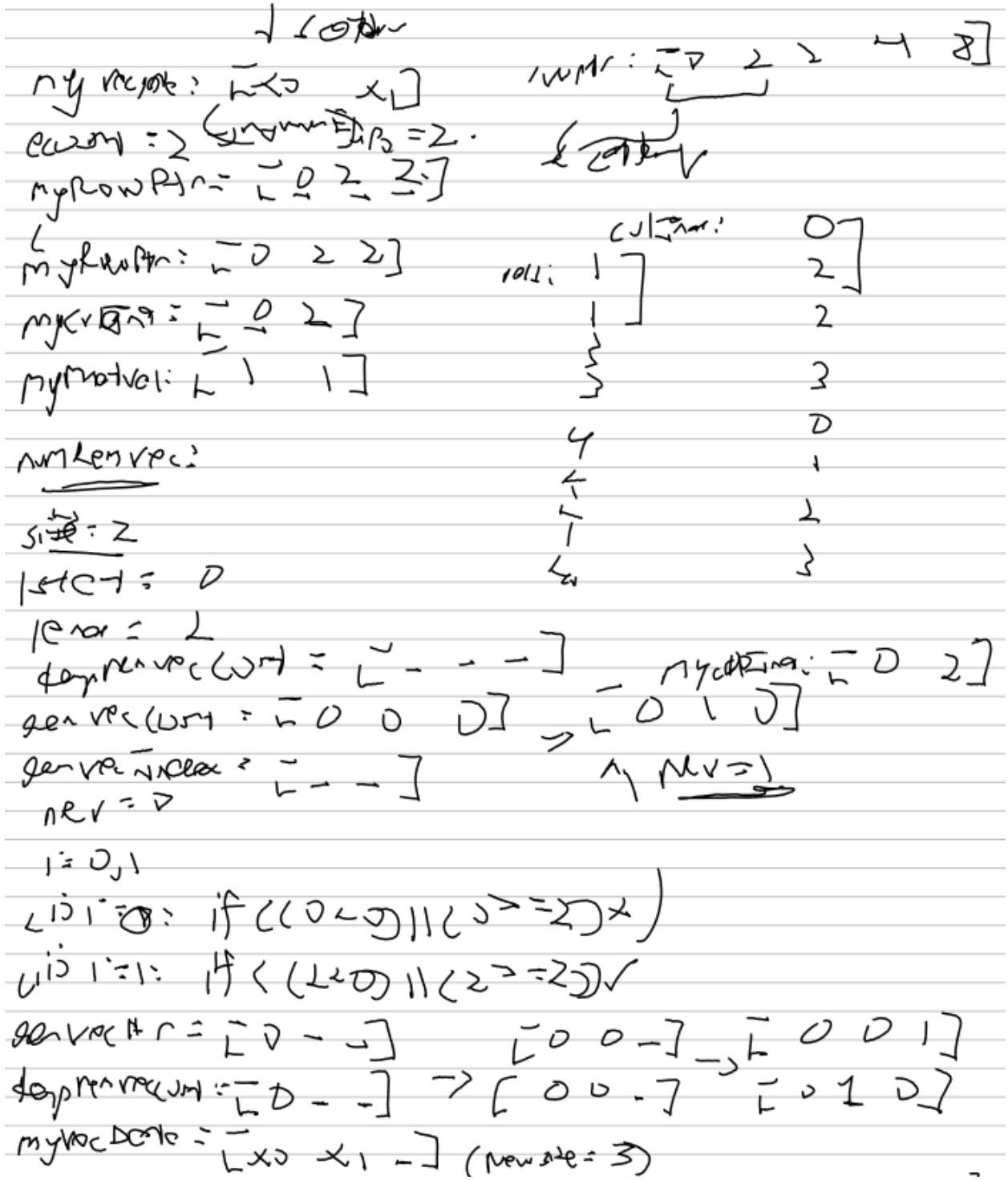
vecdata size and vecdata ptr

[2 2 0] | [0 2 4]

recvcnt and recvptr rsize = 2

[2 6 0] [0 2 8]

allvecdata: [x0, x1, x2 x3]



proc = 0
i = 0, 1

i = 0: if((0 < 0) || (0 >= 2)) x

i = 1: if((2 < 0) || (2 >= 2)) ✓

proc = 1

resultvecindex[0 + 0] = 2

resultvecindex[0] = 2

mycol[1] = 2 + 0 + 0 = 2

myvalbring: [0 2]

[0 2]

[0 2]

ARV = 1

ARV = 1.

mycolumn = [0 2]

recvdispl = [0 0 1]

resultvecindex = [2 _].

countveccount = [0 1 0]

myvecdone = [x0 x1 _]

P2:

vecdata.size and vecdata.disp

[2 2 0] | [0 2 4]

ecount and edisp

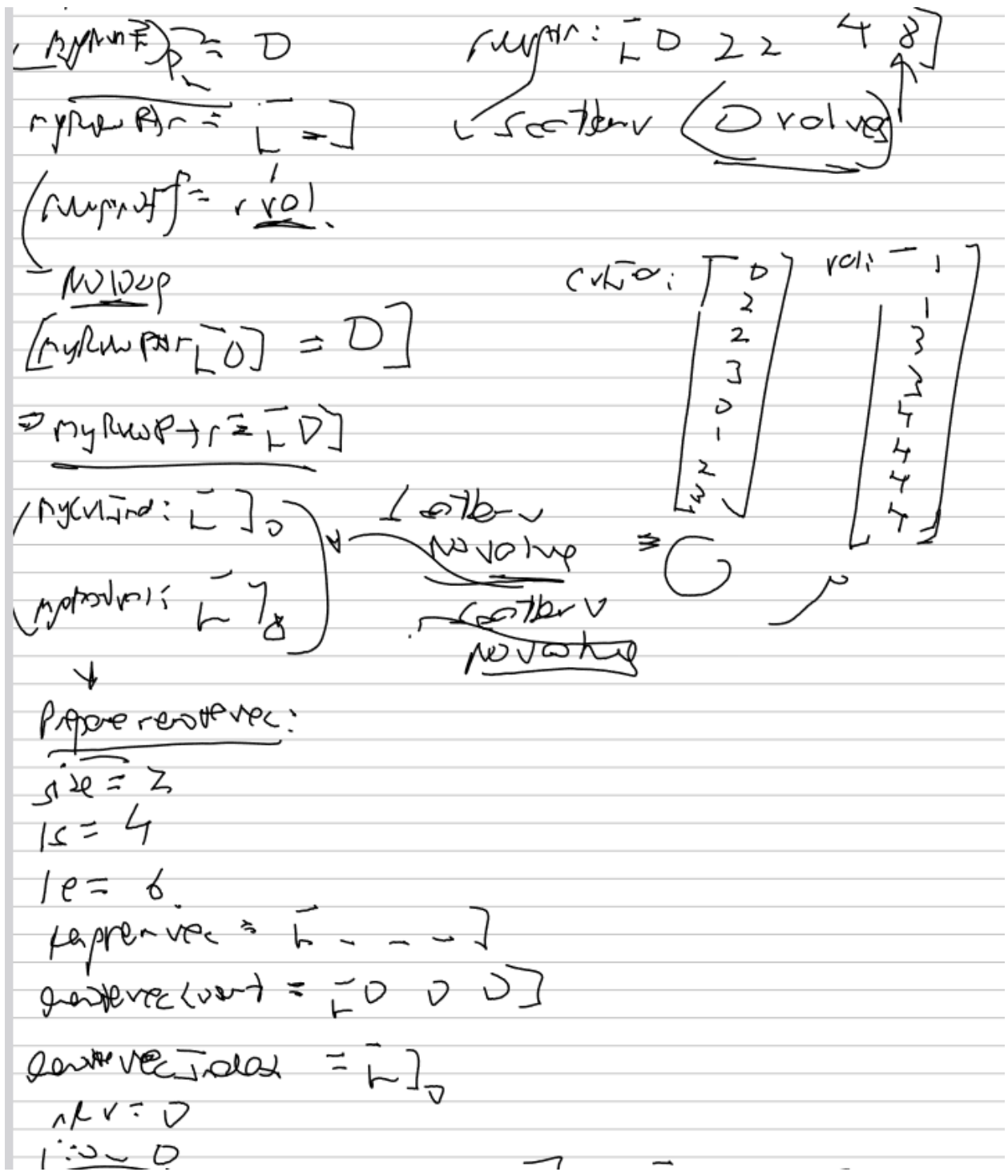
[2 6 0] [0 2 8]

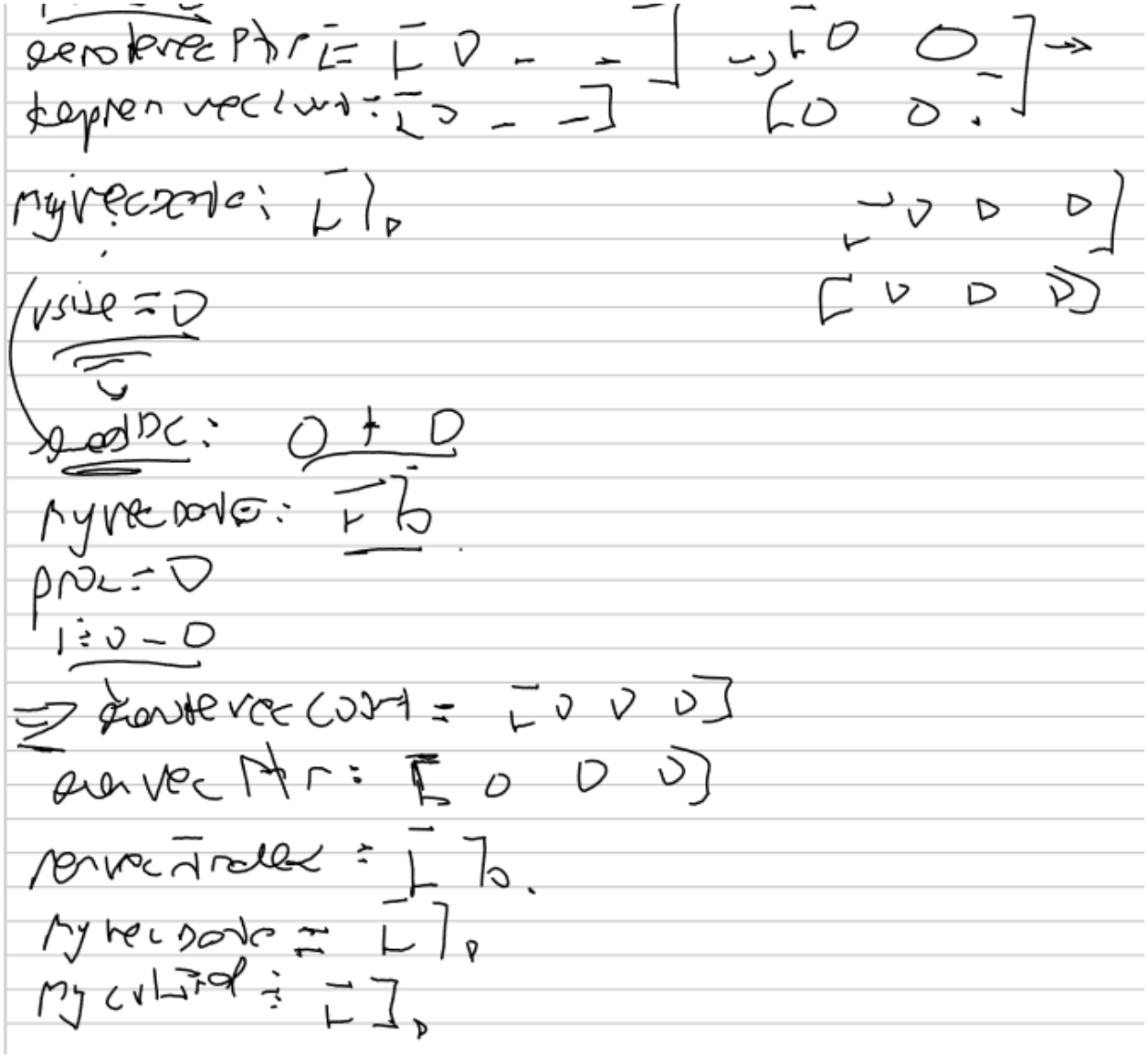
vsize = ~~x~~ = 0

size = 2

myvecdone = [ ]0

Allvecdata: [x0 x1 x2 x3]





P0: result: [_ _]

vsize = 2, myRowPtr: [0 2 2]

myColInd: [0 2]

myVal: [1 1]

myRecNode: [x_0 x_1 x_2]

⇒ i: 0; i < 2; i++

(i) i = 0: res[0] → j = 0 to 1.

res[0] = val[0] · vec[col[0]] + val[1] · vec[col[1]]

⇒ res[0] = 1 · vec[0] + 1 · vec[2]

res[0] = $1 \cdot x_0 + 1 \cdot x_2$ — (I)

(ii) i = 1:

res[1] → j = 2 to 2 ×

∴ res[1] = 0 — (II).

P1: result = [_ _]   vsize: 2, myRowPtr: [0 2 6]

myColInd: [0 1 2 3 0 1]   myVal: [3 3 4 4 4 4]
(indices 0 1 2 3 4 5)

myVec: [x_2 x_3 x_0 x_1].

i = 0 to 1.   count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

(i) i = 0: res[0] : j = 0, 1.

res[0] = val[0] · vec[colInd[0]] + val[1] · vec[colInd[1]]

(ii) i = 1: j = 2 to 5. (2, 3, 4, 5)

res[1] = val[2] · vec[col[2]] + val[3] · vec[col[3]] + val[4] · vec[col[4]] + val[5] · vec[col[5]]

res[0] = 3 · vec[0] + 3 · vec[1] = $3x_2 + 3x_3$. — (I)

res[1] = 4 · vec[2] + 4 · vec[3] + 4 · vec[0] + 4 · vec[1]

= $4x_0 + 4x_1 + 4x_2 + 4x_3$. — (II)

⇒ P0 res: $\begin{bmatrix} 1x_0 + 1x_2 \\ 0 \end{bmatrix}$ P1 res: $\begin{bmatrix} 3x_2 + 3x_3 \\ 4(x_0 + \cdots x_3) \end{bmatrix}$ P2 res: [ ] (empty)

recvDispl: [0 2 4]   recvCountsize: [2 2 0]

$\begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} \begin{bmatrix} x_0 + x_2 \\ 0 \\ 3x_2 + 3x_3 \\ 4(x_0 + \cdots x_3) \end{bmatrix}$ (Total result) //.