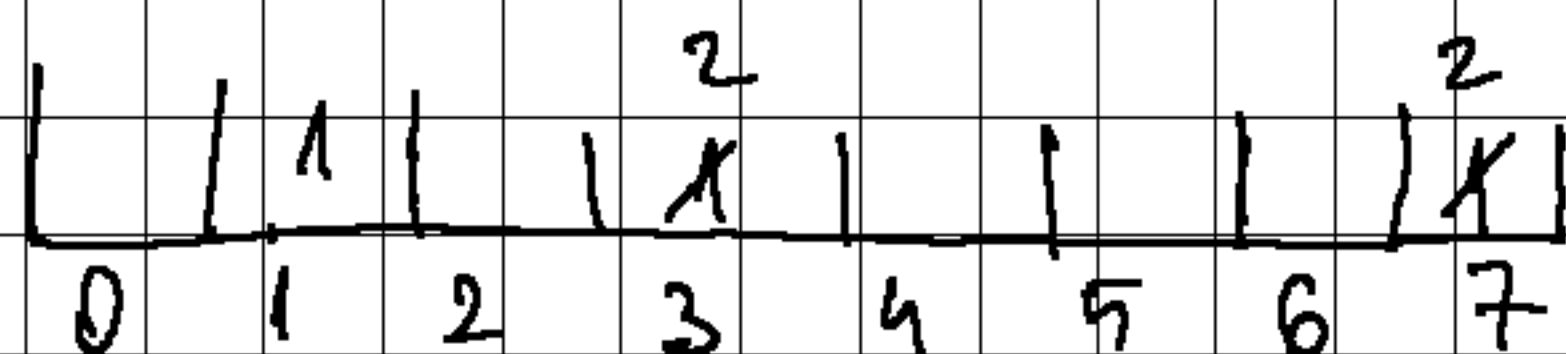


Count sort: $T(m, N) = m + N$

Ex: 7, 1, 3, 7, 3, 7

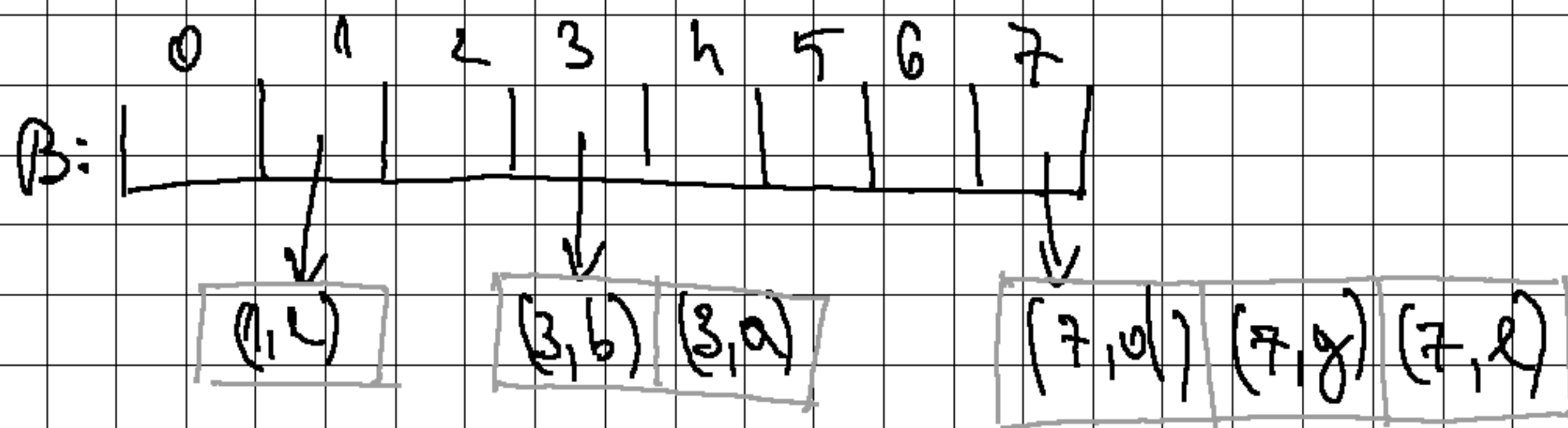


Bucket Sort:

Given: a sequence S of pairs (key, value)
keys are integers between 0 and $N-1$

Sort S :

Ex: (7, d) (1, e) (3, b) (7, g) (3, a) (7, l) $n=6$
 $N=8$



DT: Sequence

init(S)

isEmpty(S) \Rightarrow T/F

first(S) \Rightarrow e

removeFirst(S)

addLast(S, e)

Subalg BucketSort (S, N)

① allocate B . an array with N positions
each position is a sequence

for $i \leftarrow 0, N-1$ execute
 init($B[i]$)
endfor

while not isEmpty(S) execute

Stablesort \rightarrow $\langle k, v \rangle \leftarrow \text{first}(S)$
 removeFirst(S)

 addLast($B[k], \langle k, v \rangle$)

endwhile

for $i \leftarrow 0, N-1$ execute

 while not isEmpty($B[i]$) execute

$\langle k, v \rangle \leftarrow \text{first}(B[i])$

 removeFirst($B[i]$)

 addLast($S, \langle k, v \rangle$)

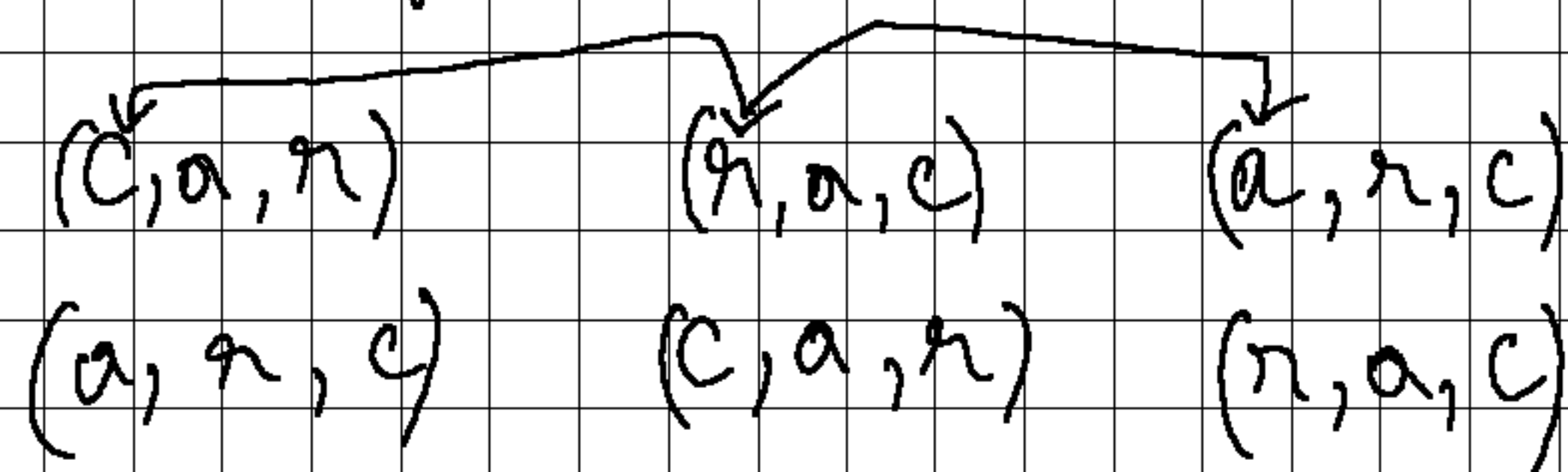
 endwhile

endfor

endsubalg

$\Theta(N + m)$

Lexicographic sort:



d-tuples

$$(x_1, x_2, \dots, x_d) \leq (y_1, y_2, \dots, y_d) \Leftrightarrow$$

$$\stackrel{\text{def}}{\Leftrightarrow} (x_1 < y_1) \text{ OR } (x_1 = y_1) \text{ AND } (x_2, \dots, x_d) \leq (y_2, \dots, y_d)$$

$$R = (R_1, \dots, R_d)$$

$$\forall i \quad x_i \text{ (R}_i\text{)} y_i$$

Subalg Lexicographic Sort (S, d, R)

for $i \in d, 1, -1$ execute
stableSort (S, R_i)
endfor
endSubalg

ex: $d=3$

$(7, 4, 6)$ $(5, 1, 5)$ $(2, 4, 6)$ $(2, 1, 4)$ $(3, 2, 4)$

Sort based on dim 3 \Rightarrow

$\Rightarrow (2, 1, \underline{4})$ $(3, 2, \underline{4})$ $(5, 1, \underline{5})$ $(7, 4, \underline{6})$ $(2, 4, \underline{6})$

Sort on dim 2 $\Rightarrow (2, \underline{1}, 4)$ $(5, \underline{1}, 5)$ $(3, \underline{2}, 4)$ $(7, \underline{4}, 6)$ $(2, \underline{4}, 6)$

Sort on dim 1 $\Rightarrow (\underline{2}, 1, 4)$ $(\underline{2}, 4, 6)$ $(\underline{3}, 2, 4)$ $(\underline{5}, 1, 5)$ $(\underline{7}, 4, 6)$

if we sort starting from dim 1 we won't get the correct result.

Merge for a sorted singly linked list.

SL List:

head: \uparrow SL Node

SL Node:

info: T Comp

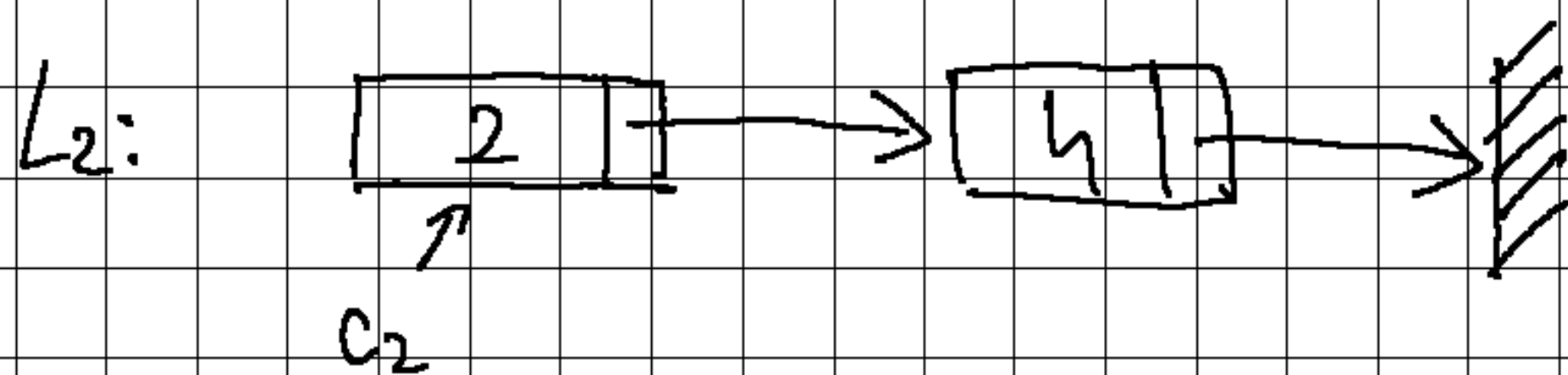
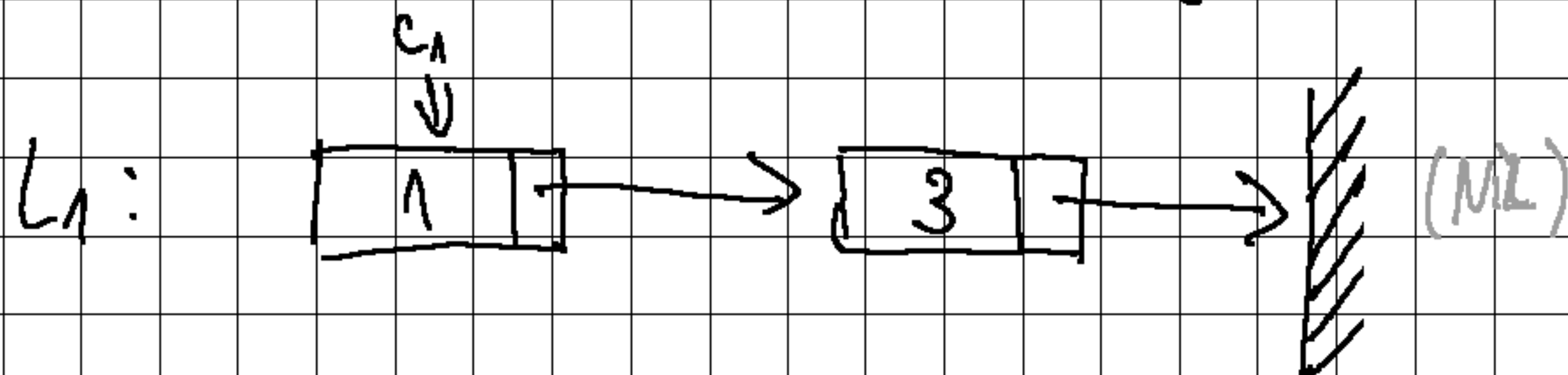
next: \uparrow SL Node

$L_1: m$
 $L_2: m$
↓
 $\Theta(m+m)$

Version 1: Copy all the elements in the result list

Version 2: Remake the links between original nodes

original list will be destroyed



LR:

headLR
tailLR

Subalgorithm merge(L_1, L_2, LR)

$C_1 \leftarrow L_1.\text{head}$

$C_2 \leftarrow L_2.\text{head}$

$\text{head } LR \leftarrow \text{NIL}$

$\text{tail } LR \leftarrow \text{NIL}$

while ($C_1 \neq \text{NIL}$) AND ($C_2 \neq \text{NIL}$) execute

if ($[C_1].\text{info} \leq [C_2].\text{info}$) then

selected $\leftarrow C_1$

$C_1 \leftarrow [C_1].\text{next}$

else

selected $\leftarrow C_2$

$C_2 \leftarrow [C_2].\text{next}$

endif

$[\text{selected}].\text{next} \leftarrow \text{NIL}$

if ($\text{head } LR = \text{NIL}$) then

$\text{head } LR \leftarrow \text{selected}$

$\text{tail } LR \leftarrow \text{selected}$

else

$[\text{tail } LR].\text{next} \leftarrow \text{selected}$

$\text{tail } LR \leftarrow \text{selected}$

endif

end while

```

- If tailLR  $\neq$  Nil then
  - If C1  $\neq$  Nil then
    [tailLR].next = C1
  else
    [tailLR].next = C2
  endif
else
  - If C1  $\neq$  Nil then
    headLR  $\leftarrow$  C1
  else
    headLR  $\leftarrow$  C2
  endif
endif
LR.head  $\leftarrow$  headLR
End Subalgorithm

```

BC: $\Theta(\min(m, n))$

WC: $\Theta(m+n)$