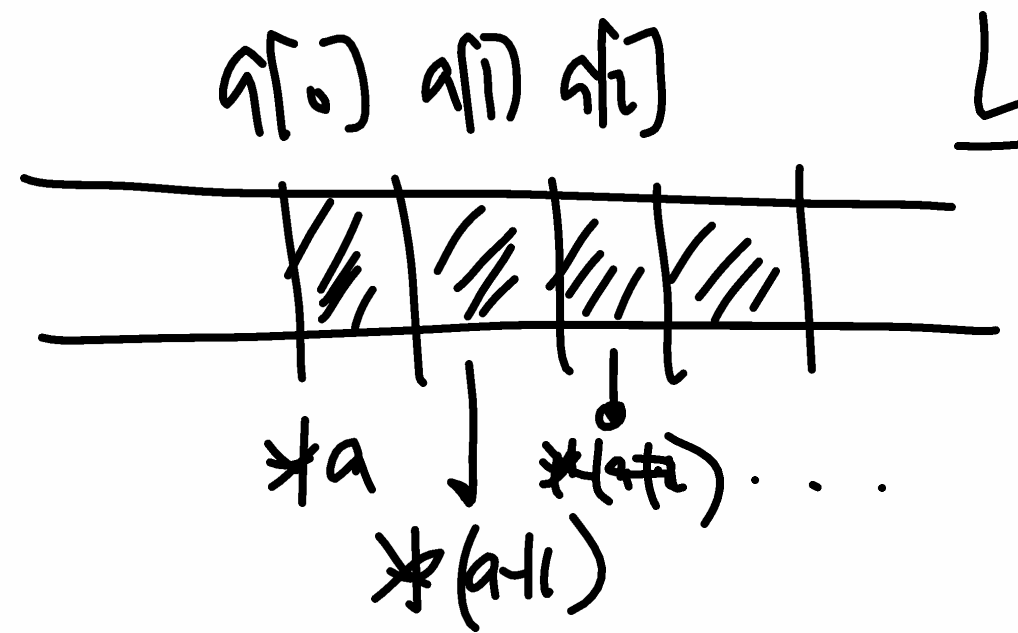


LINKED LIST (LL)

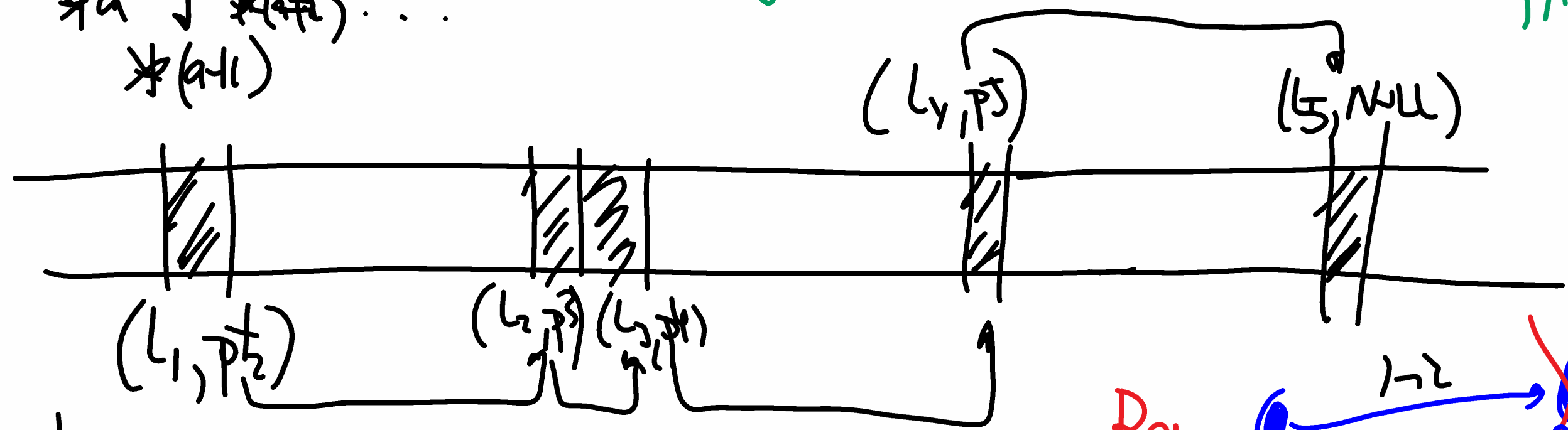
Thw 4 APR 2019

Array: a



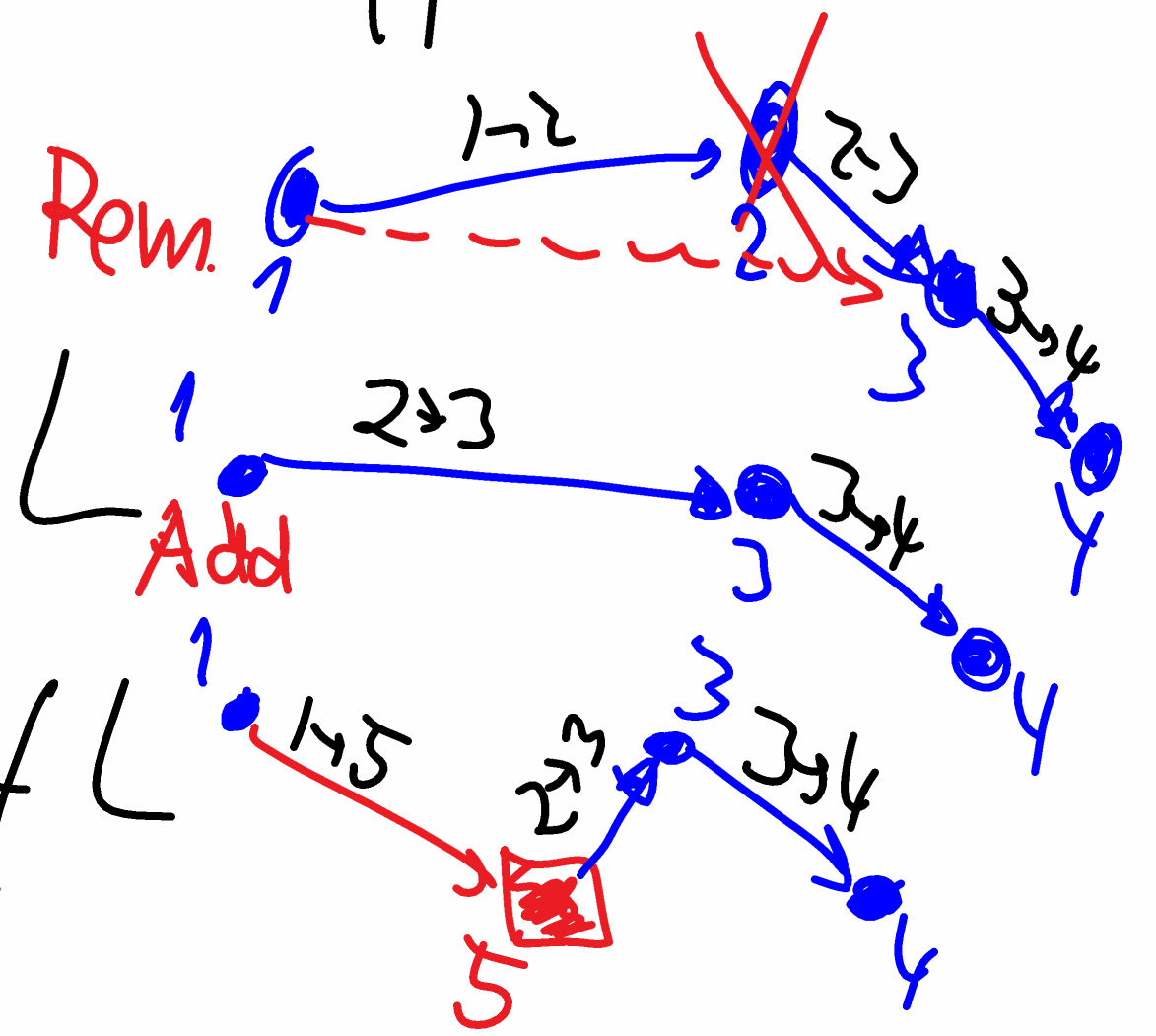
- Add/Remove elements in Array $\Rightarrow$ New Mem alloc. Hence memory inefficient!!
- Find j -th element: $a[j-1]$

Linked list: L



We want to know how to
 memory efficient
 (not time efficient)
 Highly Costly

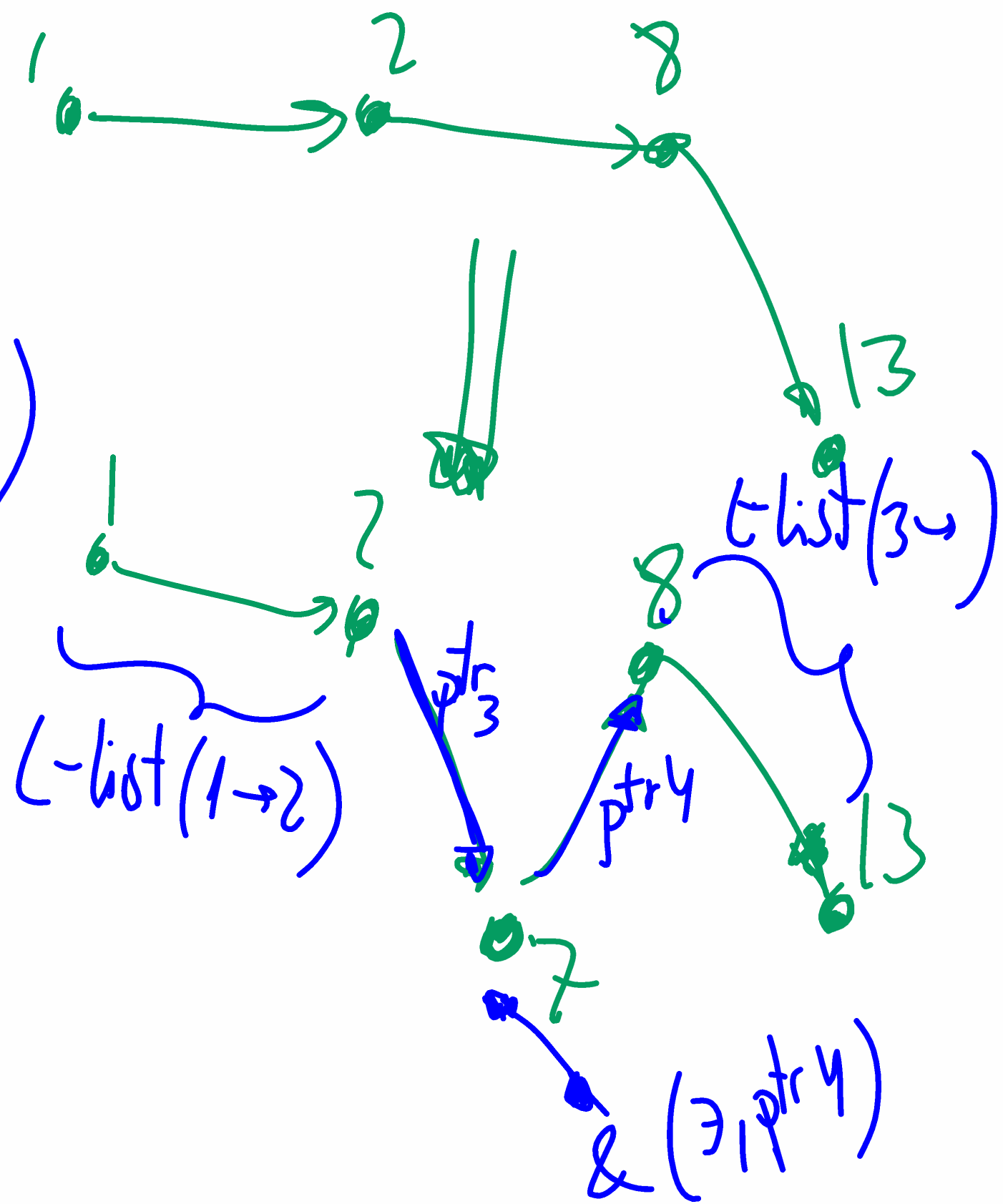
- Add/Remove elements of L
- Get/Find an element of L



L-list(4+1, 7, 3) =

= L-list(1 → 2) → ptr₃ = 8 (7, ptr₄)

~~linked list = {1, 2, 8, 13}~~



Base case: We have a l-list w/ no elements.

What's that!? It's just a NULL pointer!

$L = \text{NULL} \#$

How we add say element 17?

(Clearly this will be inserting 17 as new element #1!

$L\text{-list}(0+1, 17, 1) = (17, \text{NULL})$

HOMEWORK:

Try to implement an insert method for linked list.

If you get stuck, have a look at slides
web site for file 2019 04 04_100600-linkedlist.c