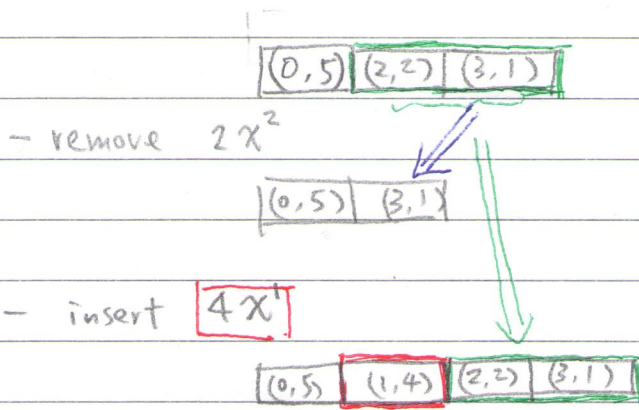


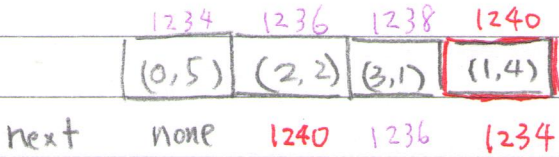
### Linked List

\* sparse polynomial w/ dense index array  
 $x^3 + 2x^2 + 5$

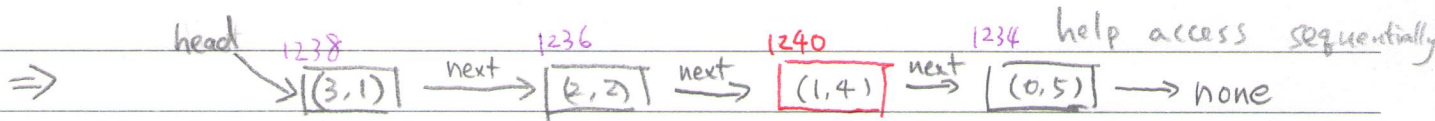


need "moving" elements

\* if "no moving" but still want to access sequentially

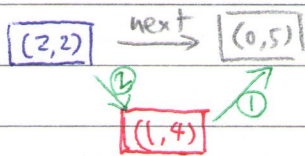


add a "next" field to



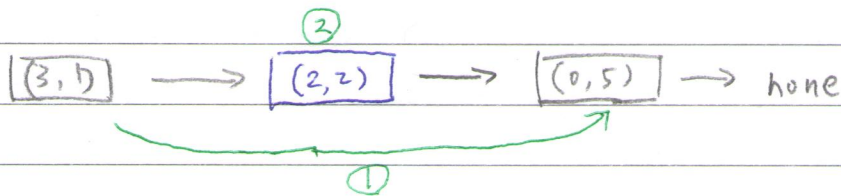
singly-linked list (chain)

\* insertion



- ①  $\boxed{(1,4)}.next = \boxed{(2,2)}.next$
- ②  $\boxed{(2,2)}.next = \text{location of } \boxed{(1,4)}$

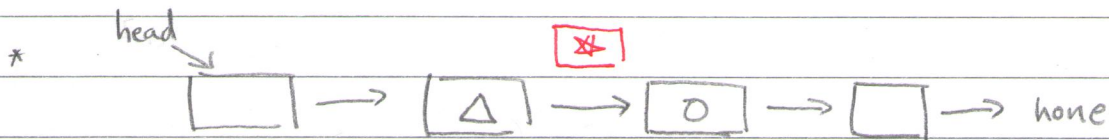
\* deletion



①  $(3,1).next = (0,5).next$

② free  $(2,2)$  (delete)

\* insert Front, removeFront (head) : see text book



- singly : insertAfter( $\Delta$ ) is easy

$\star$ .next =  $\Delta$ .next

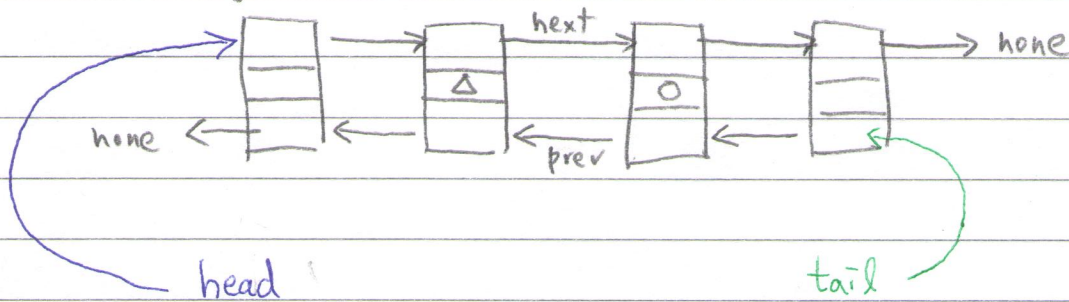
$\Delta$ .next = location of  $\star$

insert Before (0) is hard

① find  $[?]$  such that  $([?].next \text{ is } [0])$

② insert After ( $[?]$ )

- doubly linked list



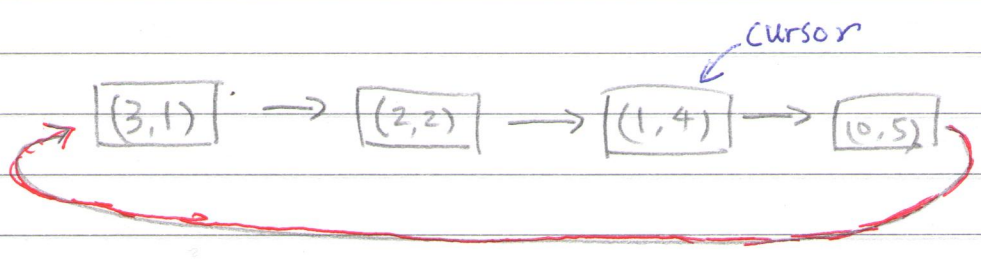
trade-off

more pointers (storage), more expensive to maintain



\* circularly linked list

"repeat all" on your MP3 player



"why?" (find out in your HW3)

\* corresponding STL: list (faster insertion/erase than vector)  
slower (no "position" access)