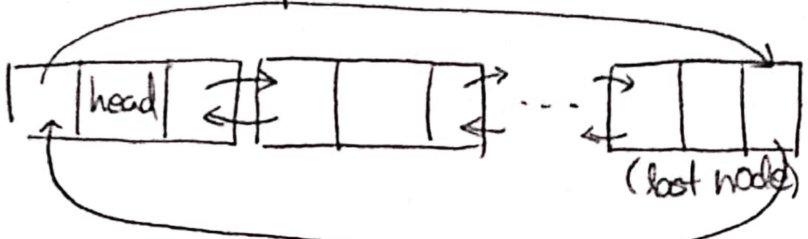
2071035
Lee Somin

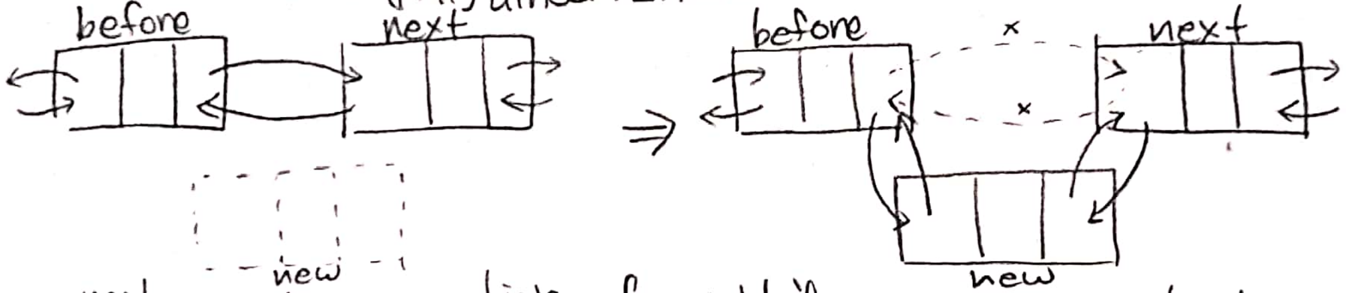
Problem 1.

- Form of Doubly Linked List:



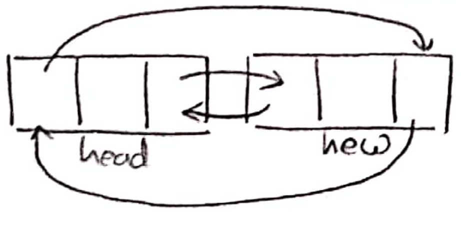
left link of head node is pointing to last node and right link of the last node is pointing to head node.

- How the Code of (p.41) dinsert_node works:



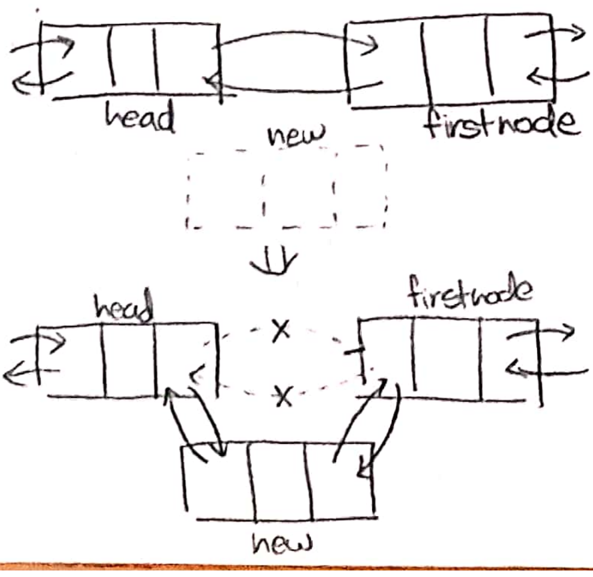
· replaces the connection of right 'from before to next' node with 'before to new' and 'new to before'.
· replaces the connection of left 'from next to before' with 'from new to next' and 'from next to new'.

* dinsert_node() when only head node exist



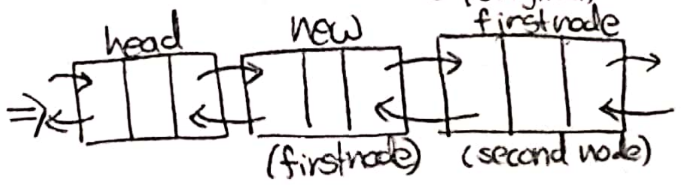
· since both links of head pointer is pointing to itself, new node and head node's (right and left) links are pointing to each other after dinsert_node is executed.
⇒ In form of doubly linked list

* dinsert_node() when inserting to the biginning.



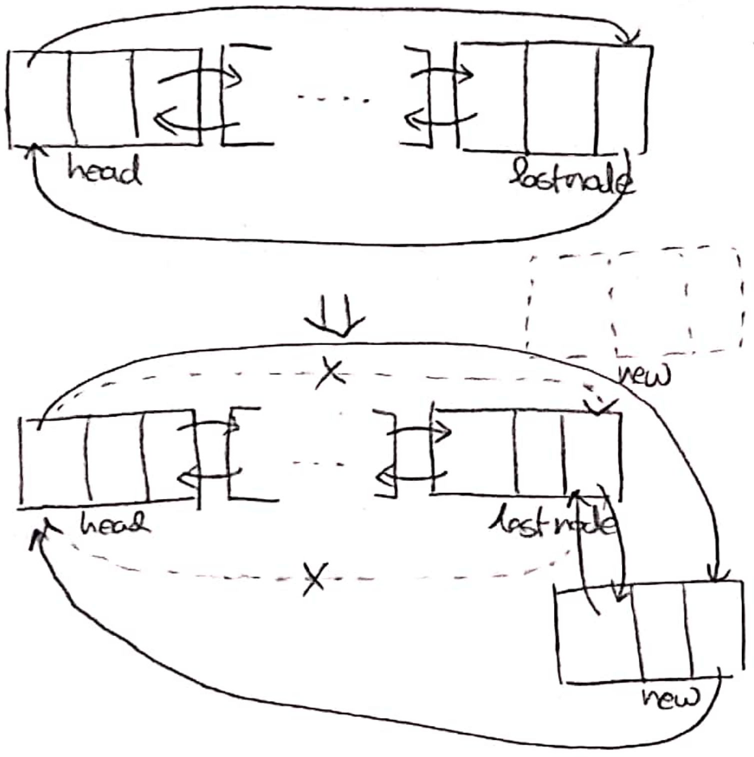
· since there is head node before the very first node of the list, the new node can be inserted between the headnode and the firstnode by dinsert_node.

⇒ In form of doubly linked list.



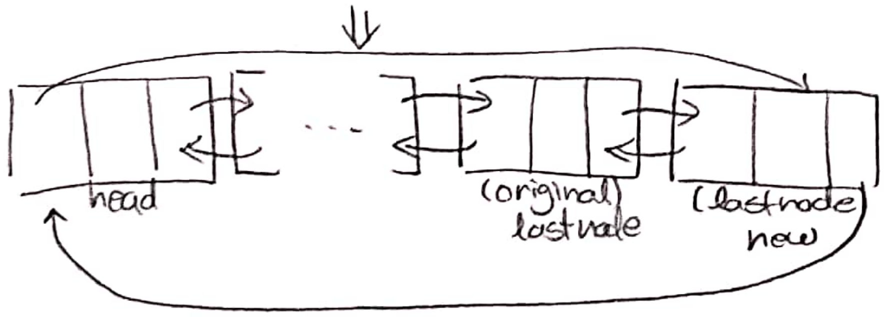
20171035
Lee Somin

*dinsert_node() when inserting at the end



- since the right link of the last node is pointing to the head node and the left link of the head node is pointing to the last node, the new node can be inserted between the original last node and the head node.
- then the new node's left link points to the original last node and the right link points to the head node.

⇒ In form of doubly linked list



* Conclusion : dinsert_node() can be used to insert a new node to the biginning or at the end of the doubly linked list.