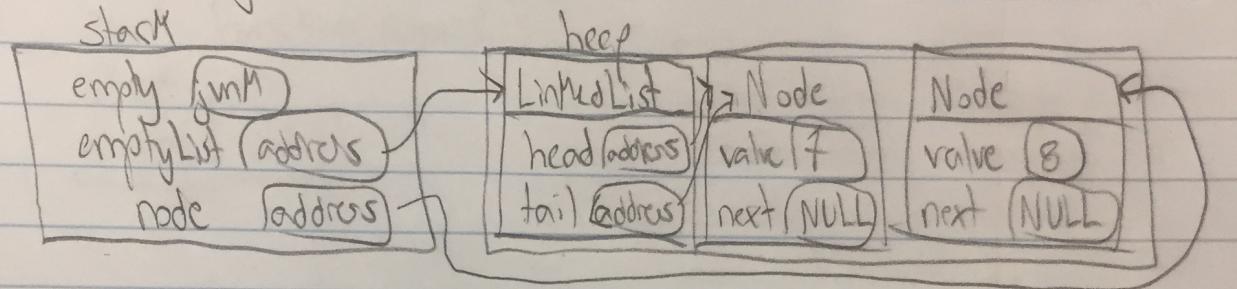
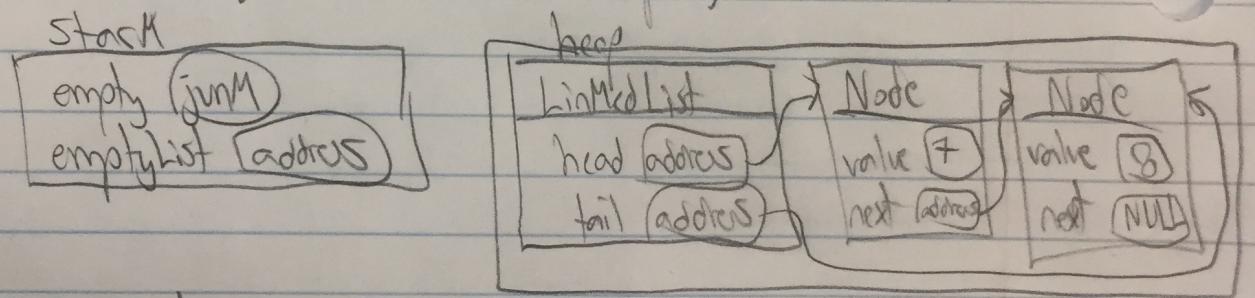


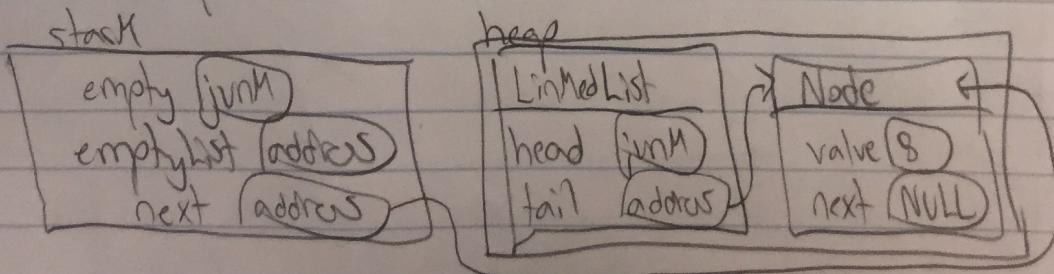
- first two lines same as Start of List Pointer Diagram
- addIntToEndOfList(emptyList, 7)
 - same as addIntToStartOfList(emptyList, 7) because code is same if head, tail = NULL
- addIntToEndOfList(emptyList, 8):
 - Node *node through node->next = NULL:



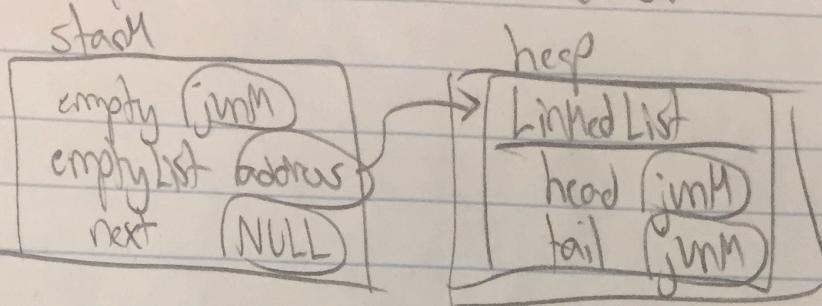
- list->tail->next = node and list->tail = node.
 - now node 7's next node points to node 8, and emptyList's tail pointer points to node 8.
- when fn returns, we lose "node" ptr, and have:



- free LinkedList(emptyList)
 - node "next" declared on stack, junk value
 - first loop iteration:
 - pointer p declared on stack, points to node 7
 - node "next" points to node 8
 - "delete p" removes node 7



'next' loop iteration removes node 8.



'delete' list gets rid of emptyList, and we lose "next" when function returns;

