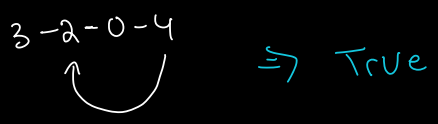
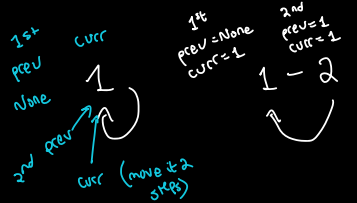
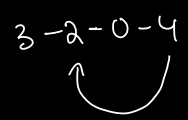
Problem: Given linked list, determine if cycle or not and determine start of cycle

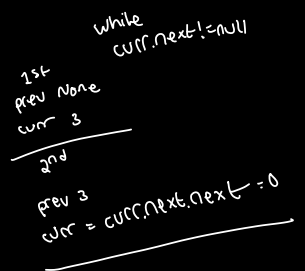


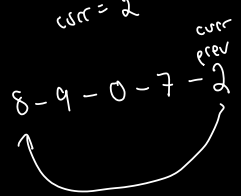
Intro:
- Verify Constraints
- Create Testcases

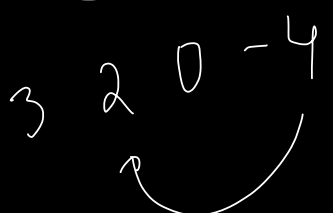
Optimal:
- Brainstorming & Pattern Observations











- Pseudocode

Technique: Tortoise & Hare
2 pointers traverse thru list, tortoise moves one step and hare moves 2 steps every iteration.
When the tortoise == hare -> cycle
Otherwise, if hare.next reaches null then there can't be a cycle

Getting the start of the loop:
Initalize 2 pointers (one at head, one at place where hare == tortoise)
iterate thru list moving each one step until (p1 == p2) and that is the start cycle node

- Write code
- Run through testcases
- Analyze time and space complexity
  - Time: O(n)
  - Space: O(1)