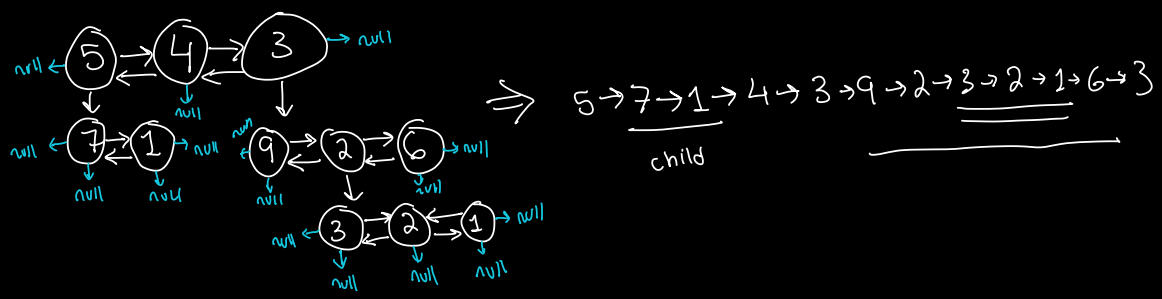
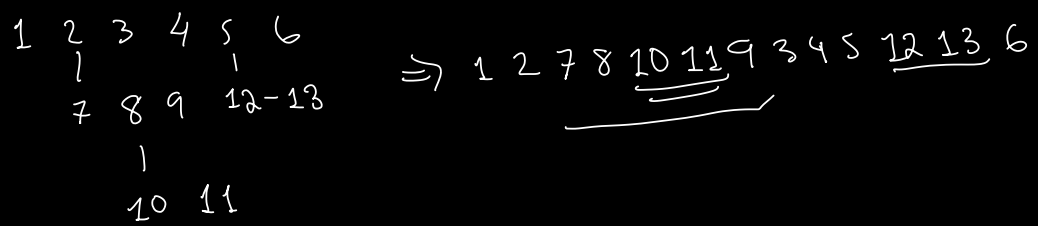
Problem: Given doubly linked list, list nodes have child property that can point to separate doubly linked list. Child lists can have own doubly linked lists. Return list as single level, flattened doubly linked list



Intro:

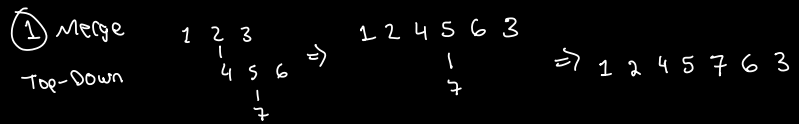
- Verify Constraints
  - every list node level can multiple children
  - child properties after flattening? set to null
- Create Testcases
  -

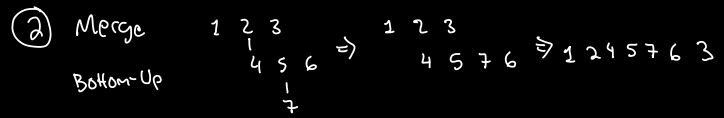


Brute Force:

- Brainstorming & Pattern Observations

seems easier



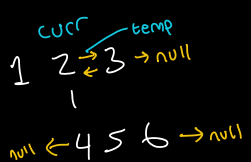


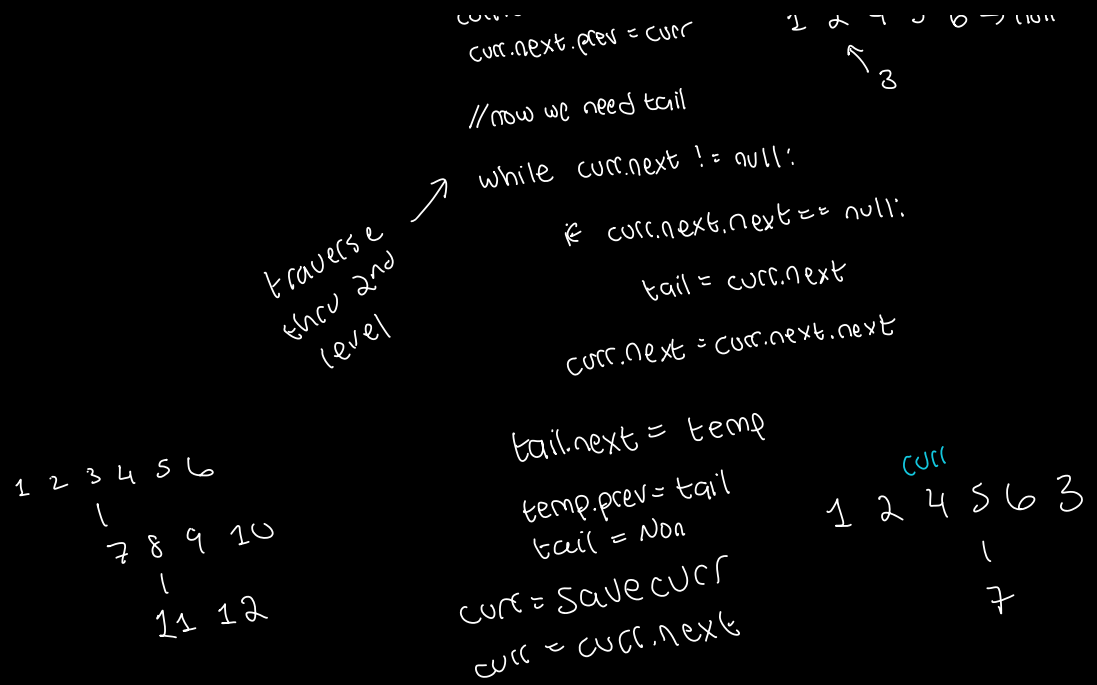
Gee

- Pseudocode

traverse thru first level

if curr.child != null :
  save curr = curr
  temp = curr.next
  curr.next = curr.child
  curr.child = null





- Write code
- Run through testcases
- Analyze time and space complexity
  Time: O(n): touch every node at least once
  Space: O(1)

Optimal:

- Brainstorming & Pattern Observations
- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity