**White board 9**

# White board 9

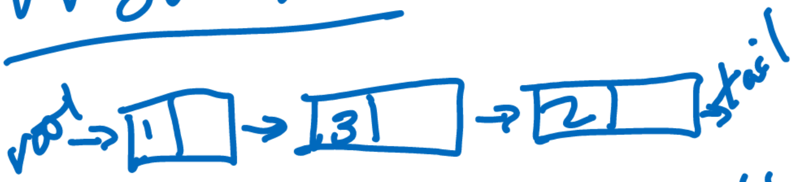
Loop Detection

## Problem Domain

Return a true/false for a method that evaluates a loop list. method takes in no arguments (make a property of linked lists class)

**Big O** required to stay at O(1) (no added memory use allowed

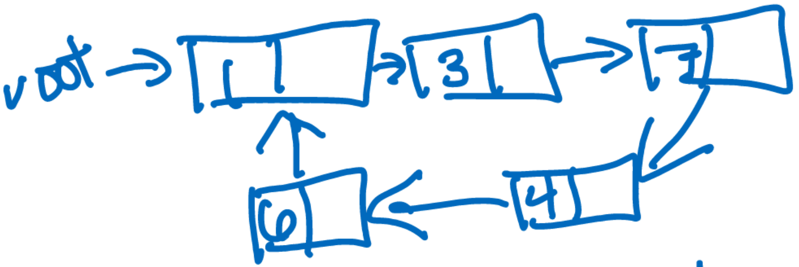
# Visual

Ⓘ 

Ⓞ false ~~no duplicates~~

∴ not a looped Linkedlist (LL)

case1

case2 Ⓘ 

Ⓞ true - looped list because end of LL points to ~~root~~ head

# Algorithm

- set a pointer to firstnode
- Set end pointer to 3rd node (+2)

```
while(node.next!=null)
if(firnode.next ≠ 3rd node)
    inext ++ (each one)
    keep going looping till null
    return false;
if (pointer A == pointer B)
    return true
```

## Code

```
public class Boolean hasLoop() {
        LinkedList testList =
                new LinkedList();

        ListNode pointerNodeA =
                new ListNode ( );
        pointerNodeA = Ø;
        pointerNodeB =
                pointerNodeA+2;
    while (pointerNodeA !null() {
            if( pointerNodeA ==
                pointerNodeB {
                return true
        }
```

```
                elseif(pointernodeA !=
                        pointernodeB) {
                      pointernodeA++;
                 }
           retun false;
      } //while loop close
  } // method close
```