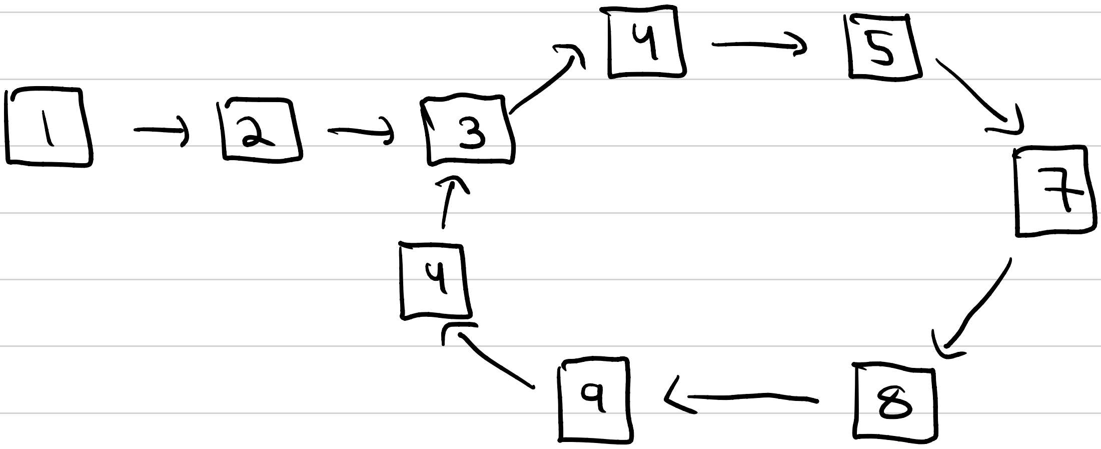
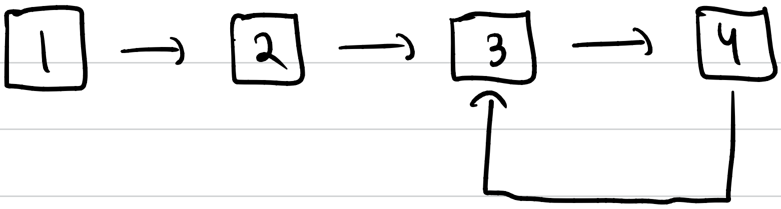
DETECTING A LOOP IN A LINKED LIST

☆ In this problem, we are given a linked list and our job is to return true or false if there is a loop in the linked list

A loop would look something like



or just.



Bruteforce solution is to use a map data structure, key is Node and value is a boolean. At each iteration, we check if the Node has a trace in the map, if it does that means we are in a loop. Time complexity is O(N) but so is the space complexity

We will reuse the Tortoise & Hare algorithm for Optimal Approach. The fast pointer at some point in time will collide with the slow pointer as there is a loop.

```
bool   detectLoop(Node * head)  {
    Node * s = head ;
    Node * f = head ;
    while(f != nullptr && f → next != nullptr){
        s = s → next ;
        f = f → next → next;
        if(s == f)  {
            return true;
        }
    }
    return false ;
}
```