Interview Question

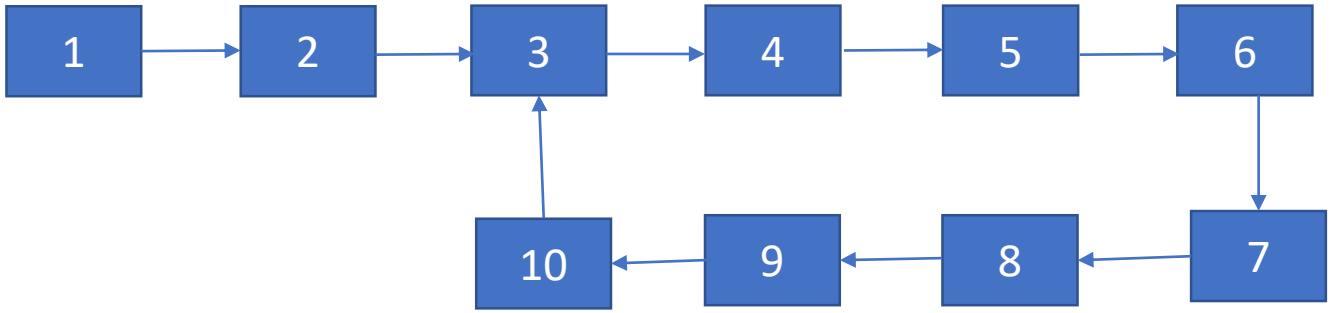
Linked list loop

- Write a function that:
  - Takes in a linked list (i.e., u pass in the head node of a linked list)
  - Returns
    - True if the linked list has a loop
    - False otherwise
  - So, it would look something like:
    - bool HasLoop(Node head)

- Look at next slide for my attempted visualization ☺

- Linked list with a loop

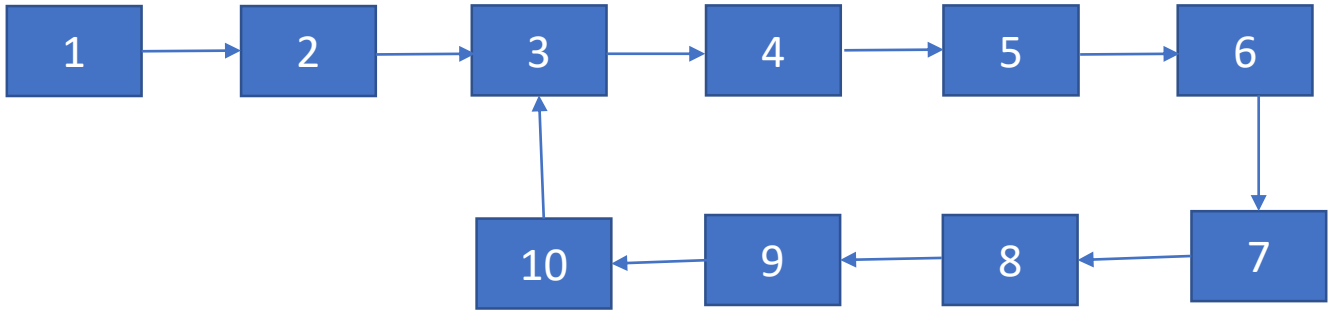
  - 1 is the head node
  - Node 10 is the last node.
  - The loop is there because Node10.next points to Node3
  - If the loop was NOT there, then Node10.next would point to NULL



- Remember:
  - u do not have to think of an efficient solution right away.
  - And u do not (SHOULD NOT) start answering right away.
    - If u do, it is **not** a sign of confidence, or intelligence, but rather a sign of immaturity in jumping to a solution or code.

  - So, first make sure u understand the question well, ask questions on the question.

- What are the various solutions that u can think of here?
- Thought 1 ← Change font color to reveal



- Thought 2 ← Change font color to reveal

- Thought 3

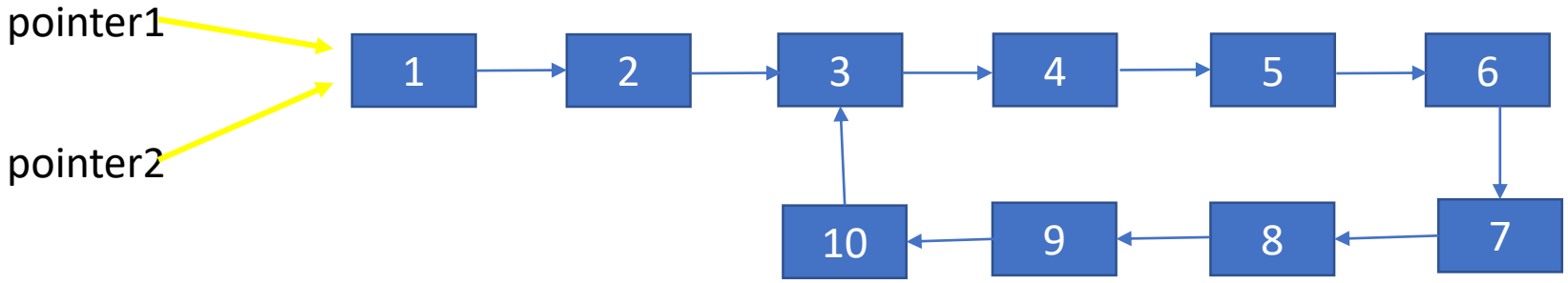
    - Mark nodes as u visit them.
    - Then if u get to a node already marked as visited, u detected the loop.
    - Cons:
        - You are modifying the nodes, which is
            - Not good.
            - Also, u may not be allowed to change the nodes, i.e., it may not be ur class, because u could just be providing this a function to be used by others.

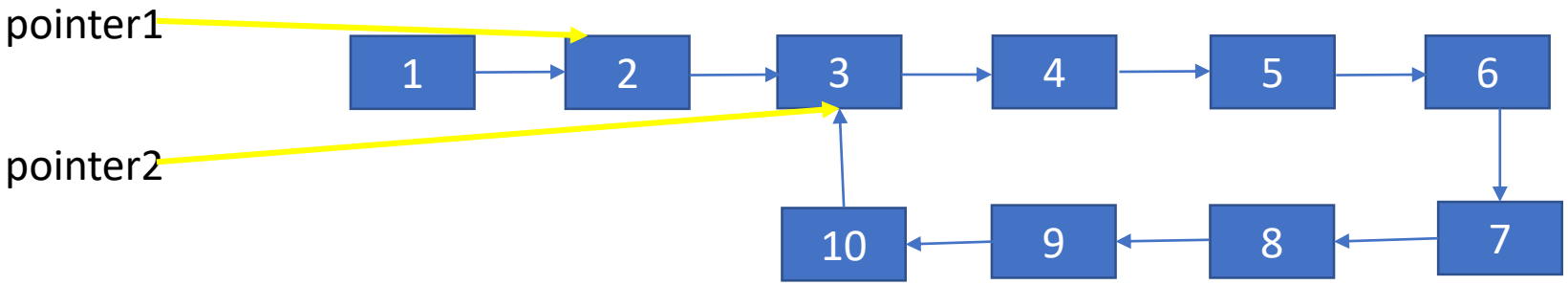
        - Complexity?
            - Time
            - Space

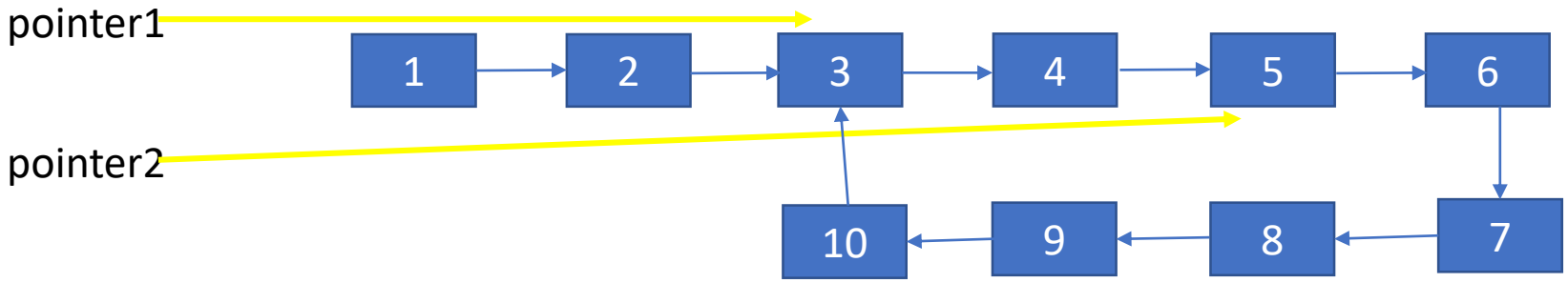
- Thought 4

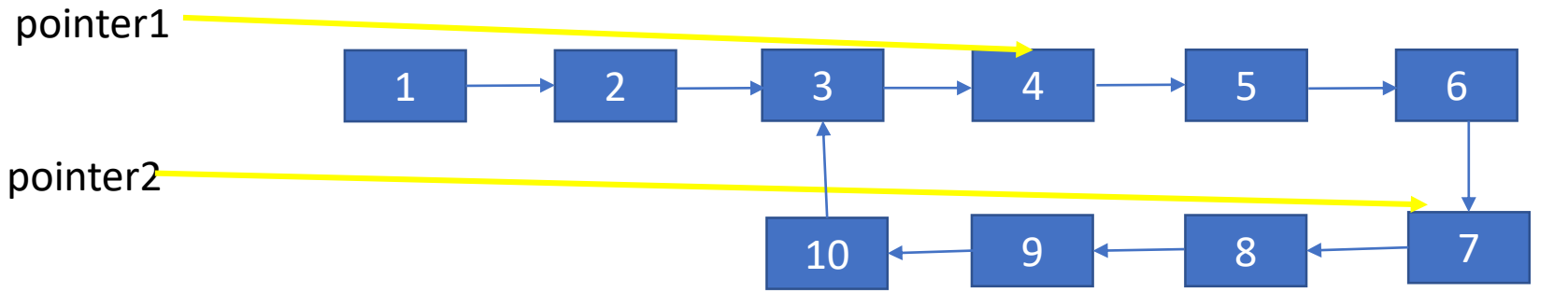
    - Have two pointers.
    - One moves along the list.
    - The 2nd pointer moves at double the speed of the 1st pointer
    - If the two pointers meet, then we have a loop
        - By pointers meeting, we mean
            - they are pointing to the same node,
            - or in other words,
            - pointer1 == pointer2

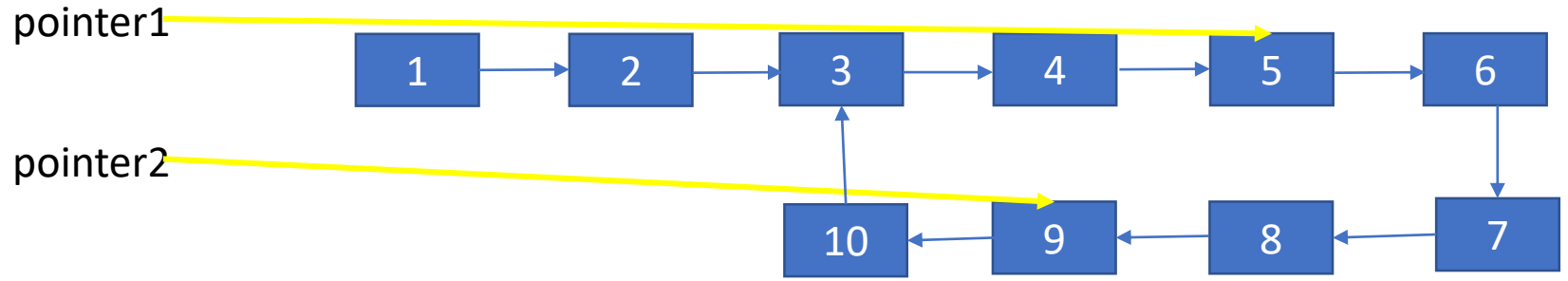
Detecting presence of a loop

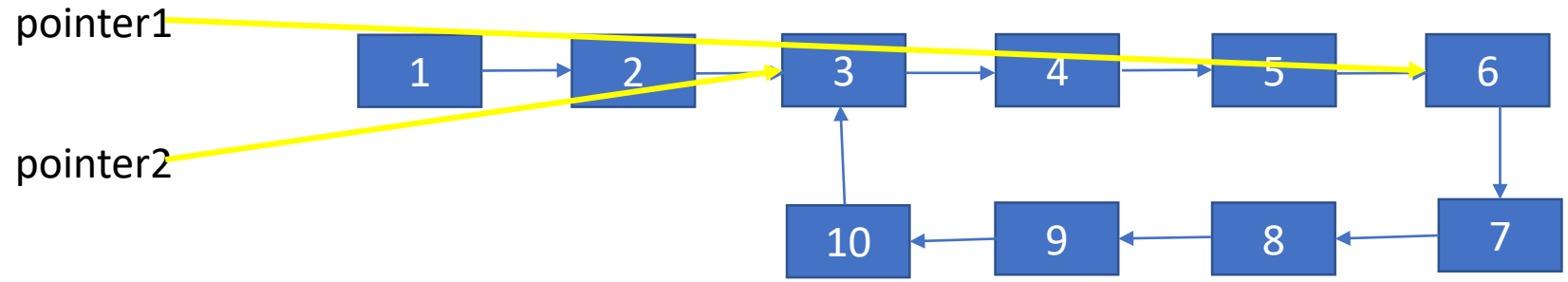


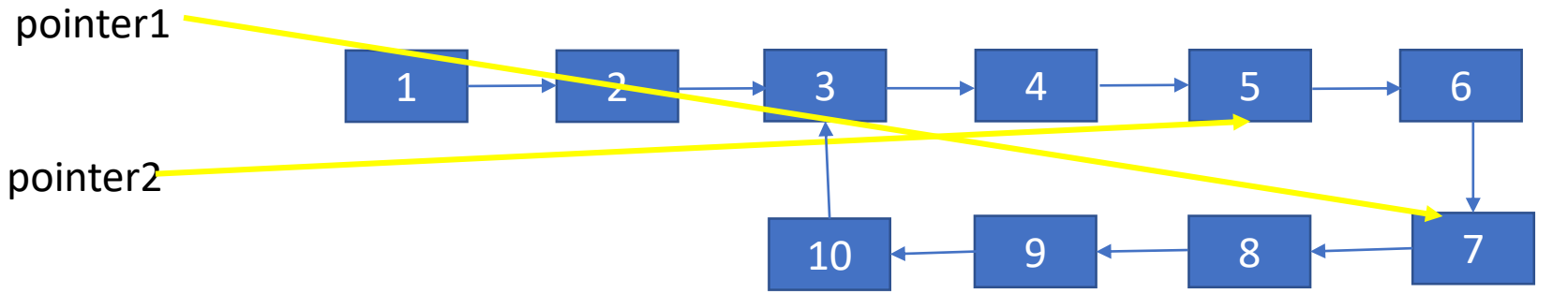


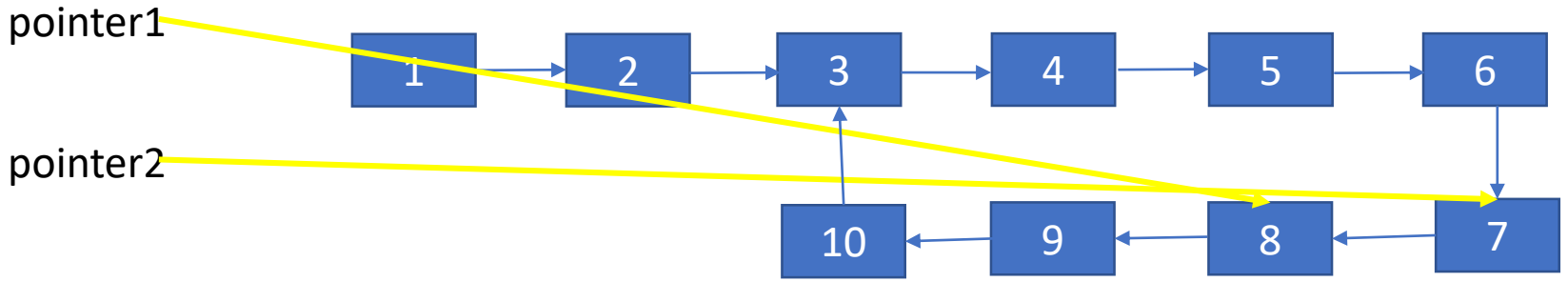


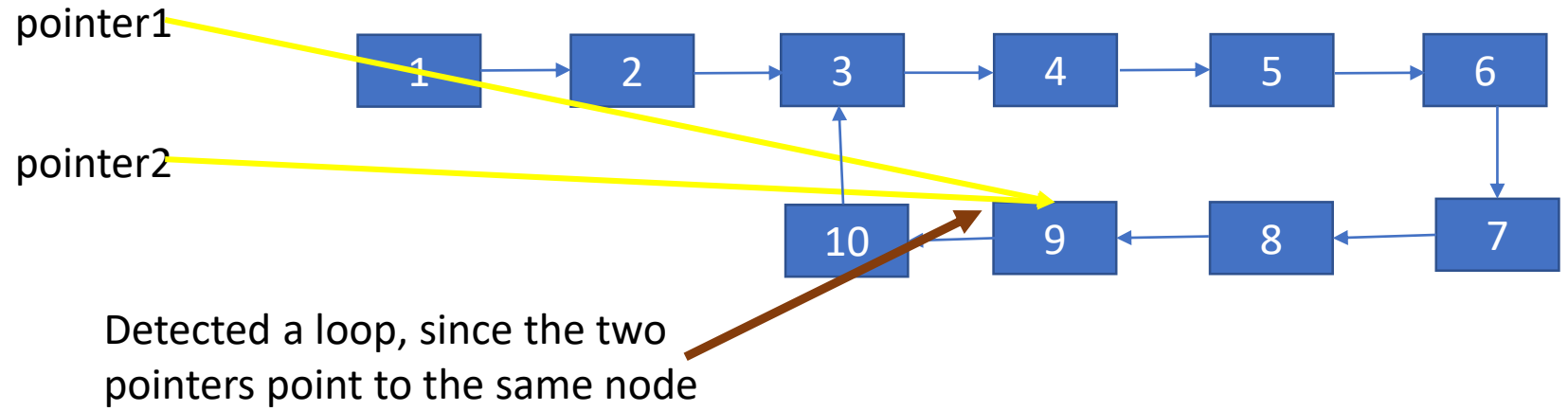

pointer1
pointer2
1
2
3
4
5
6
7
8
9
10


pointer1
pointer2
1
2
3
4
5
6
7
8
9
10


pointer1
pointer2
1
2
3
4
5
6
7
8
9
10


pointer1
pointer2
1
2
3
4
5
6
7
8
9
10


pointer1
pointer2
1
2
3
4
5
6
7
8
9
10


pointer1
pointer2
1
2
3
4
5
6
7
8
9
10
Detected a loop, since the two
pointers point to the same node

Finding start node of loop

Now that we detected a loop, we will find the starting node of the loop.

When we are at the state below (having just detected a loop), we will do the following steps:

1. Reset one pointer to head of list.
2. Now increment both pointers by 1 at a time.
3. When they meet (point at the same node), that's the starting node of the loop