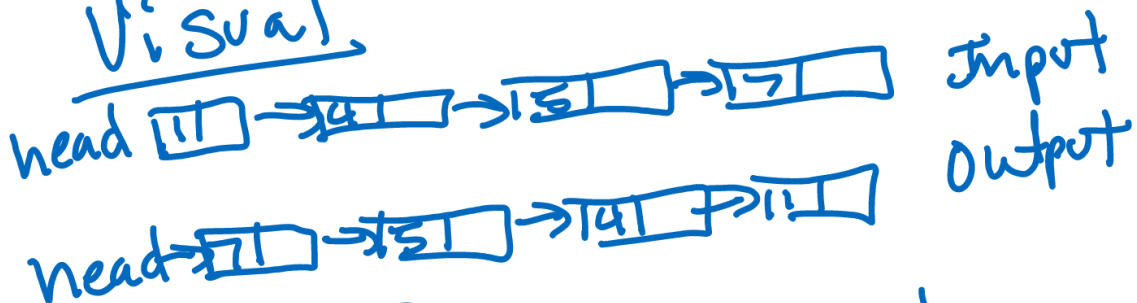


Whiteboard 10

Problem Domain

Reverse a singly linked list without using any builtin methods. Aim to have an $O(n)$ Space solution.
*note no returning.

Visual



so how?
Array.size - 1 = this.root
this.root.next = Array.size - 2
while loop? while Array.size > 0

Algorithm

- find size of Linked list using previously coded method
- Create a while loop that decrements size until its zero
- set LL root to LL to size - 1.
- go through loop till all ListNodes are relinked using next property
- return Reversed list

Big O

this solution keeps
 $O(n)$ space the same
(no new list)
and has a single while
loop so time
is also $O(n)$

Pseudo Code

ll = linkedList
int size = ll.size() - 1;
this.root = size - 1
int cycle = ll.size()
while (cycle < 0)
 assign this.root.next = size--
then return list

Code

```
public static void returnLinkedList() {
    int size = linkedList.size() - 1;
    this.root = linkedList[size];
    int cycle = linkedList.size();
    this.root.next = linkedList[size - 1];
    // now new root has new connection to
    // next node
    ListNode current = this.root.next;
    while (cycle - 2 > 0) {
        current.next = linkedList[size - 2];
        current = current.next;
        size--;
    }
    System.out.println(linkedList);
}
```