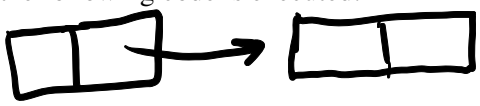
CS 61BL Summer 2021 | Small Group Tutoring Section 3: Linked Lists | Worksheet 3

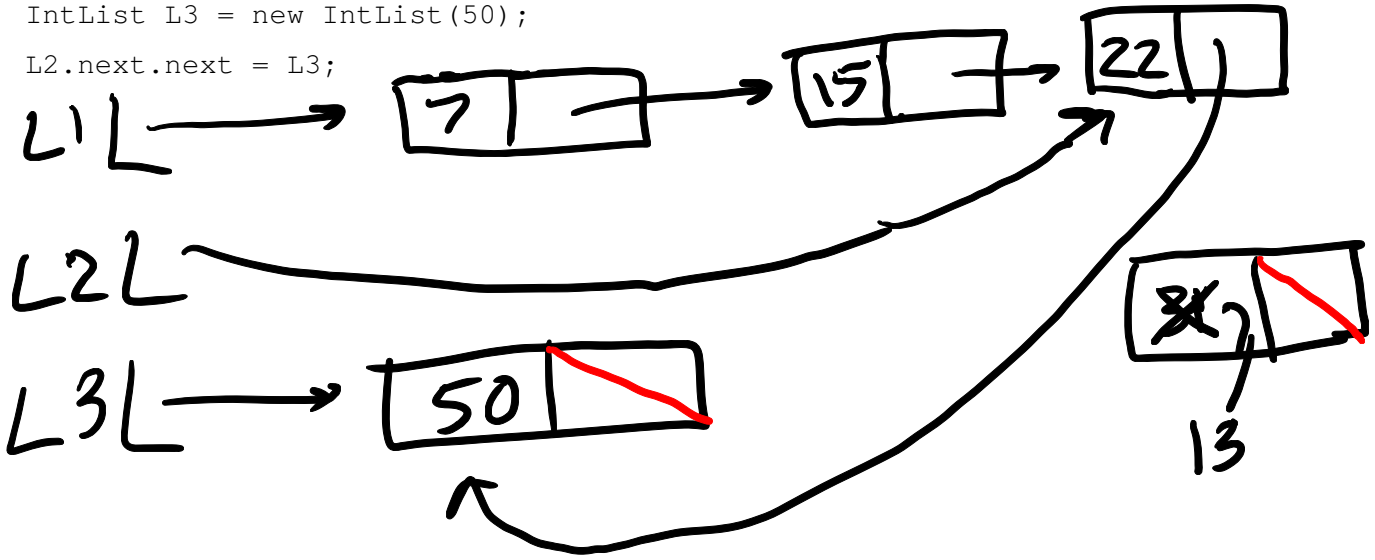
1 Pointer Practice

Draw the resulting box and pointer diagram for the IntLists after the following code is executed:



IntLists

```
IntList L1 = IntList.of(7,15,22,31);
IntList L2 = L1.next.next;
L2.next.item = 13;
L1.next.next.next = L2;
IntList L3 = new IntList(50);
L2.next.next = L3;
```



2 Skip Me

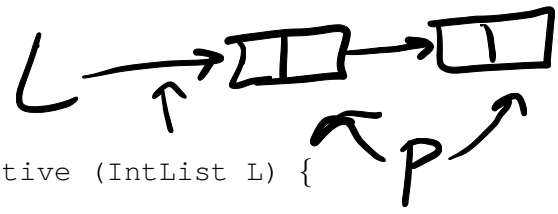
Write a function that takes in an IntList `L`, which must contain at least one element, and returns an IntList with every odd indexed element removed, starting at index 0. For example, if `L = [1, 2, 3, 4]`, the function should return an `IntList` with elements `[1, 3]`.

1. **Destructive**: `IntList L` should be modified

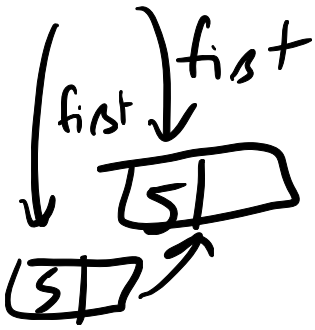
```
public static void skipDestructive (IntList L) {
    if (L == null || L.next == null) {
        return;
    }
    L.next = L.next.next;
    skipDestructive(L.next);
}
```

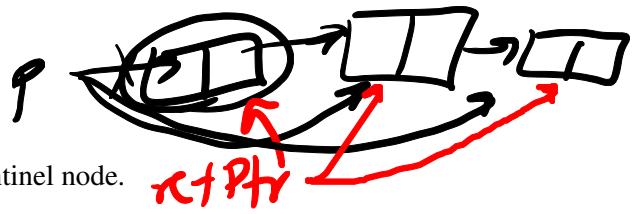
base
action / call / step

2. **Nondestructive**: `IntList L` should not be modified




```
public static IntList skipNondestructive (IntList L) {
    IntList pointer = L;
    IntList retPtr = new IntList(pointer.item);
    IntList retHead = retPtr;
    while (pointer.next != null && pointer.next.next != null) {
        retPtr.next = new IntList(pointer.next.next.item);
        pointer = pointer.next.next;
        retPtr = retPtr.next;
    }
    return retHead;
}
```

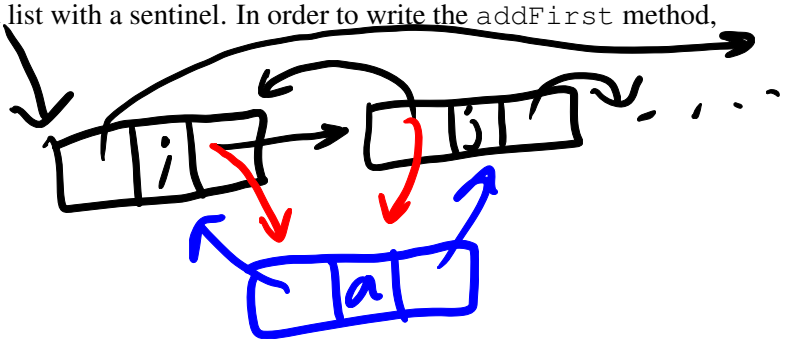




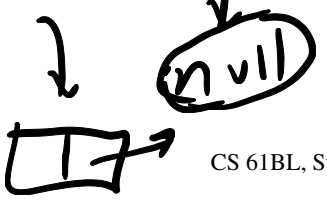
3 Benefits of Enhancements

addLast is becoming easier

1. List one advantage of having a sentinel node.
2. Suppose we implement a doubly linked list with a sentinel. In order to write the `addFirst` method, which pointers will we change?



Reason: can't reach a null pointer!



- `sentinel.next`
- `sentinel.prev`
- `sentinel.next.prev`
- `sentinel.next.next`