COMP1511 PROGRAMMING FUNDAMENTALS

Lecture 14

Linked Lists: deleting from a linked list

YESTERDAY...

- Inserting into a linked list anywhere
- Searching through the linked list for specific conditions

TODAY...

- Linked list - deleting from a linked list
  - When only one item in list
  - In the middle of list
  - At the tail

WHERE IS THE CODE?

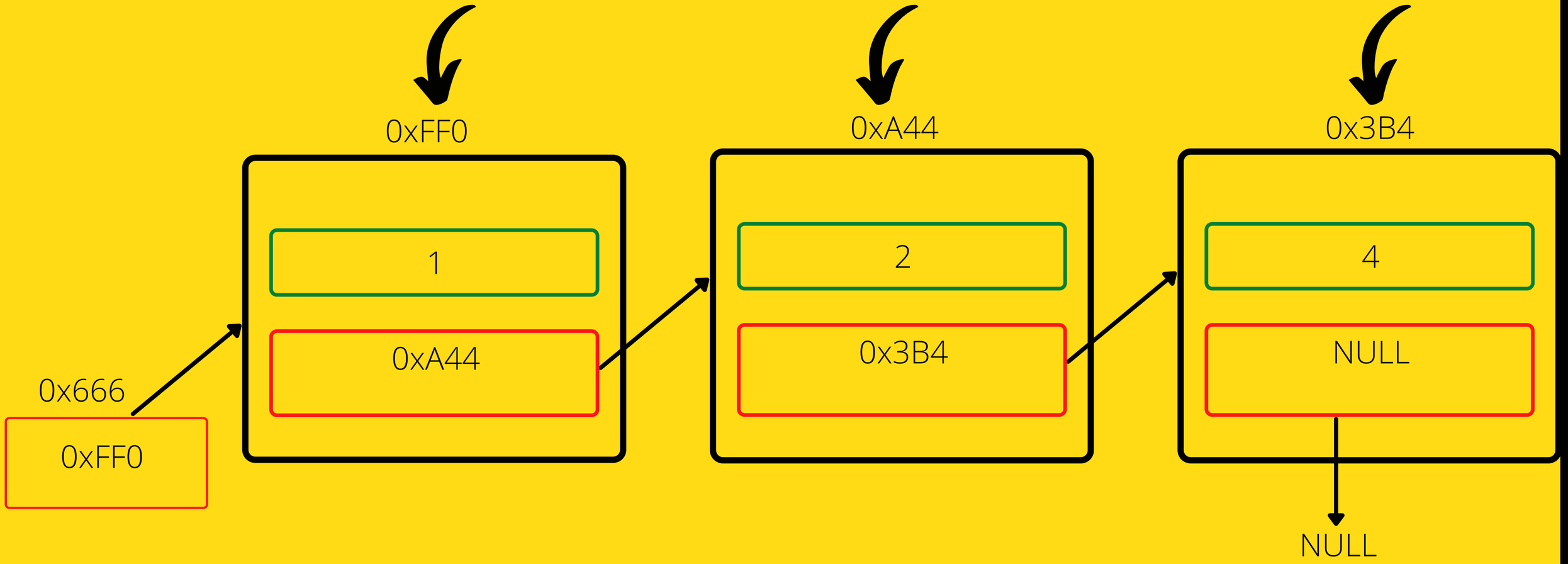
LIVE LECTURE CODE CAN BE FOUND HERE:

https://cgi.cse.unsw.edu.au/~cs1511/21T3/live/Week08/

LINKED LISTS

DELETING

- Where can i delete from a list?
  - Nowhere! (it is an empty list)
  - The head
  - Between any two nodes that exist
  - The tail (last node)



LINKED LISTS

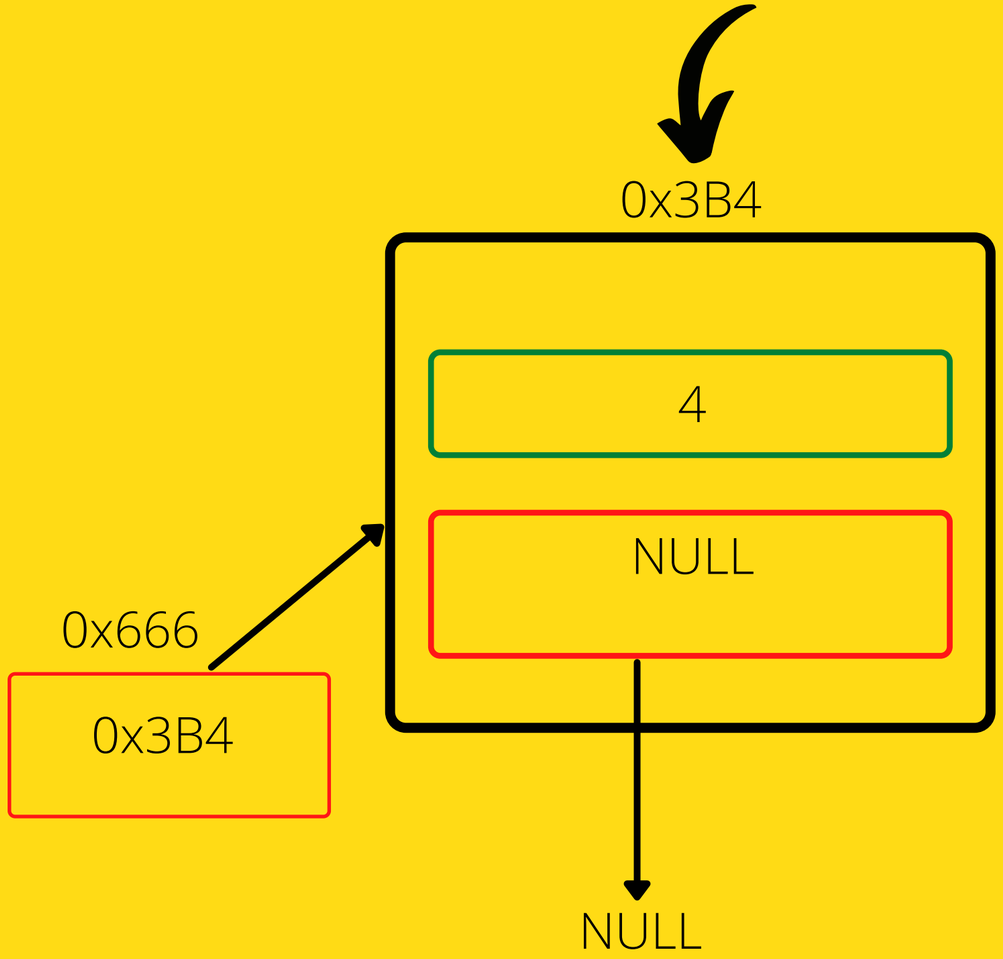
DELETING (SPECIAL CASES)

- Deleting when:
  - Nowhere! (it is an empty list)
    - Check if list is empty
    - If it is – return NULL

LINKED LISTS

DELETING (SPECIAL CASES)

- Deleting when:
  - There is only one item in the list

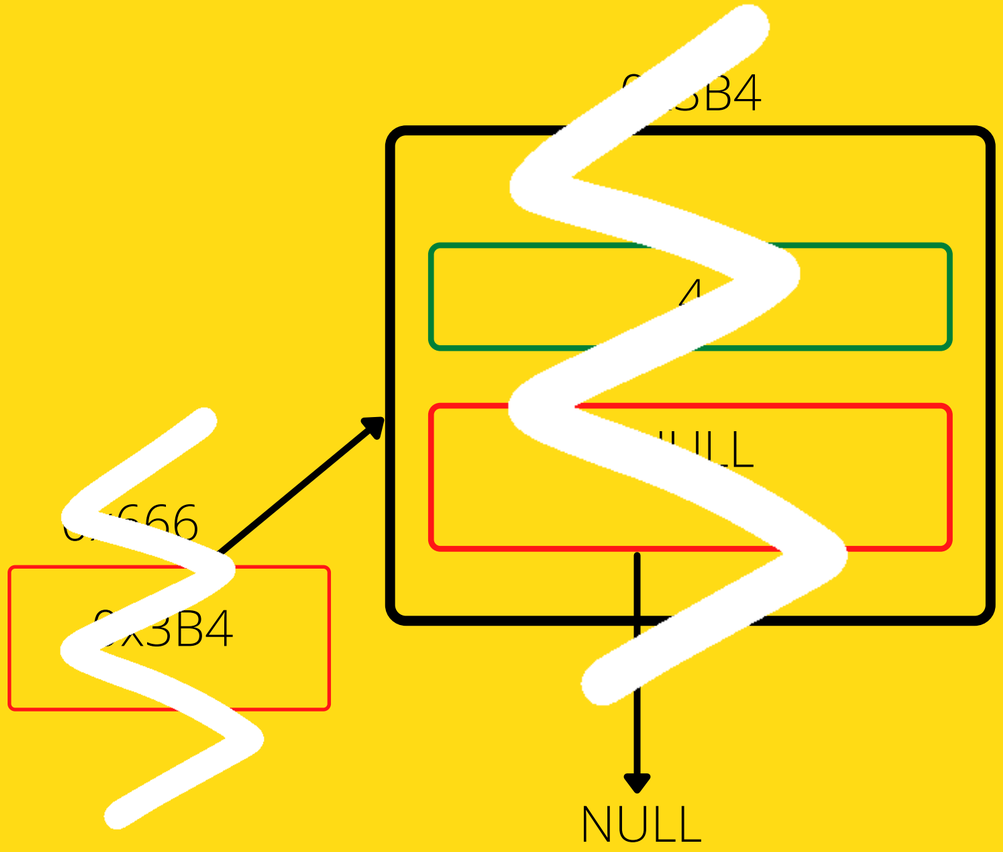


LINKED LISTS

DELETING (SPECIAL CASES)

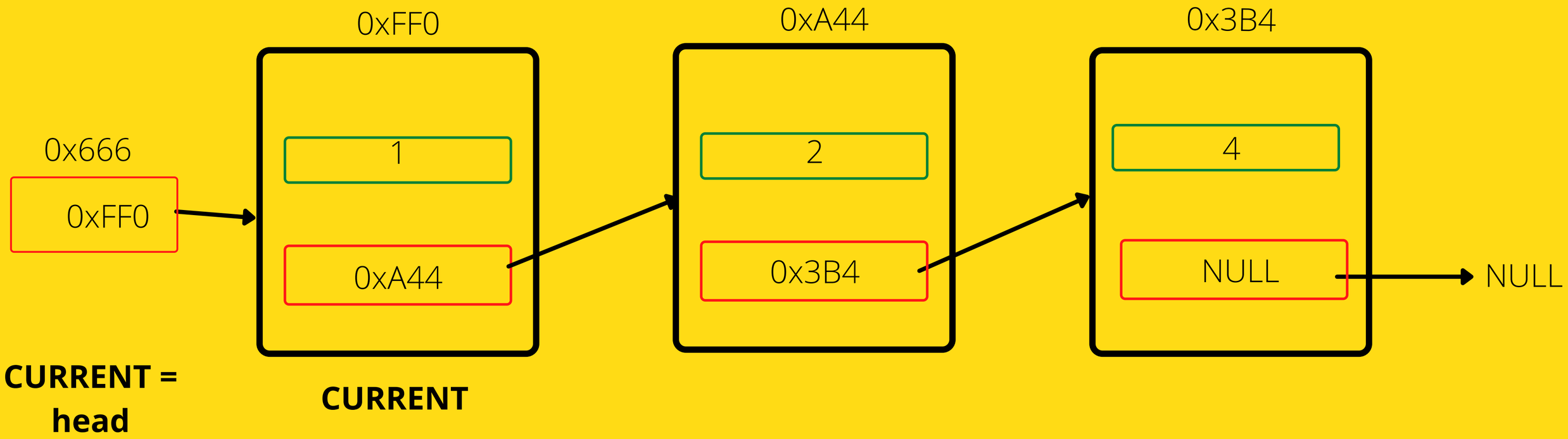
- Deleting when:
  - There is only one item in the list

1) Free the head node



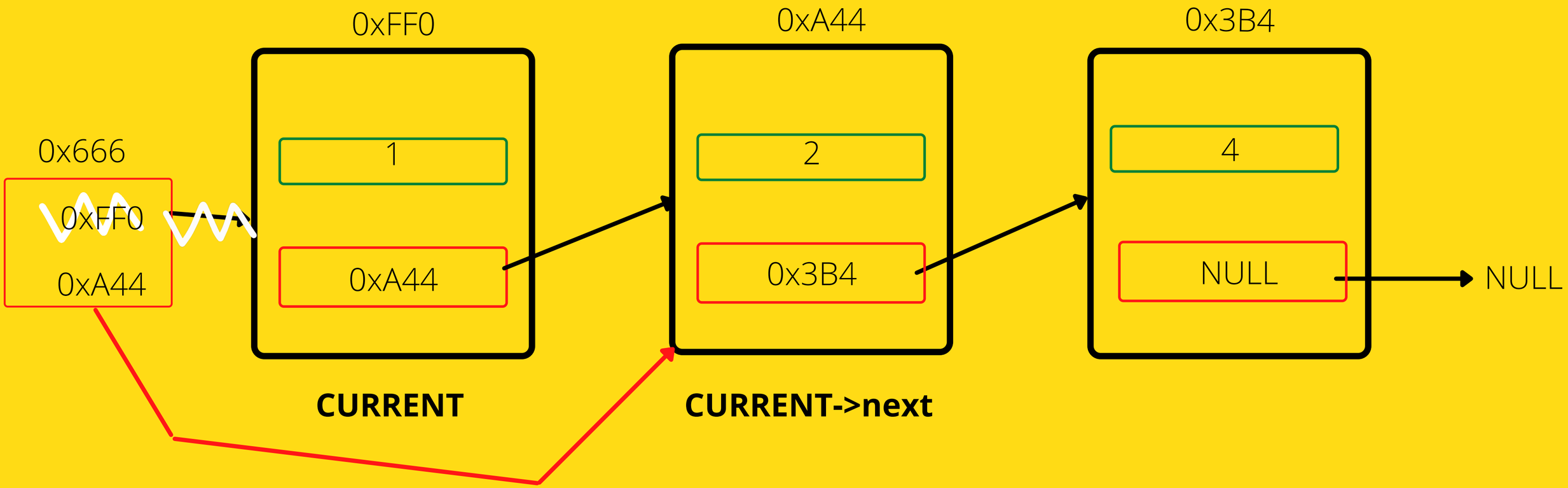
LINKED LISTS (DELETE 1 - THE HEAD OF LIST WITH OTHER ITEMS)

1. FIND THE NODE THAT YOU WANT TO DELETE



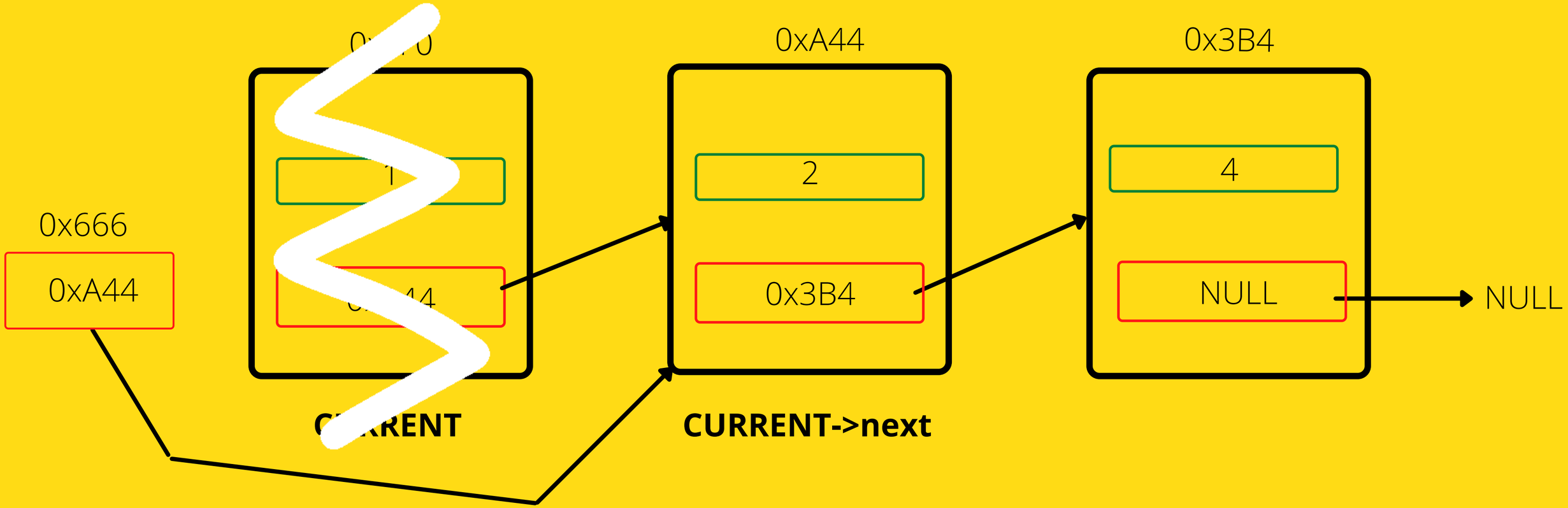
LINKED LISTS (DELETE 1 - THE HEAD OF LIST WITH OTHER ITEMS)

2. POINT THE HEAD TO THE NEXT NODE



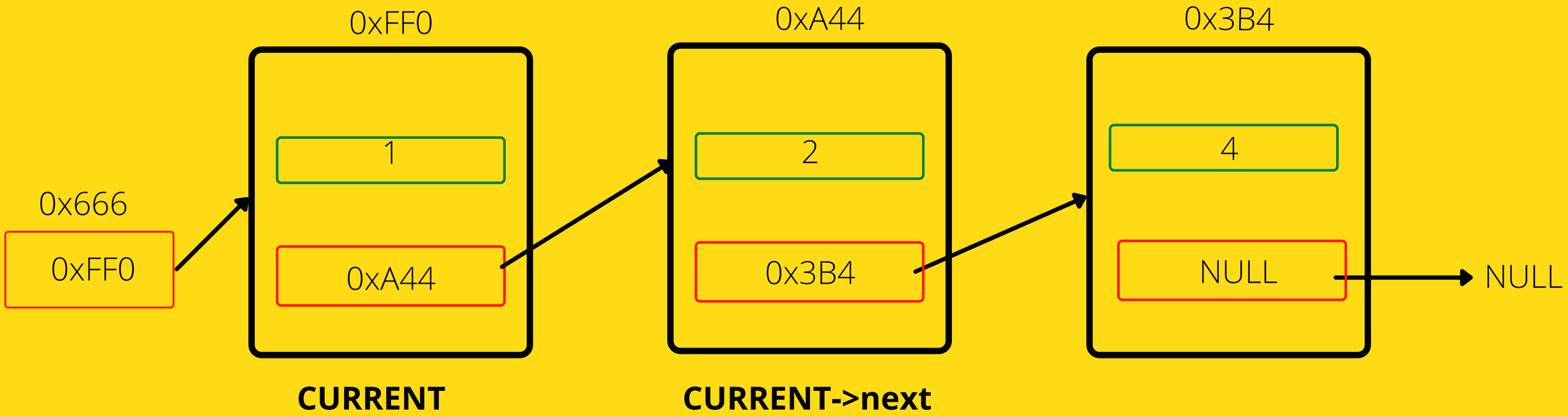
LINKED LISTS (DELETE 1 - THE HEAD OF LIST WITH OTHER ITEMS)

3. FREE THE CURRENT HEAD



LINKED LISTS (DELETE 2 - IN THE MIDDLE OF TWO NODES)

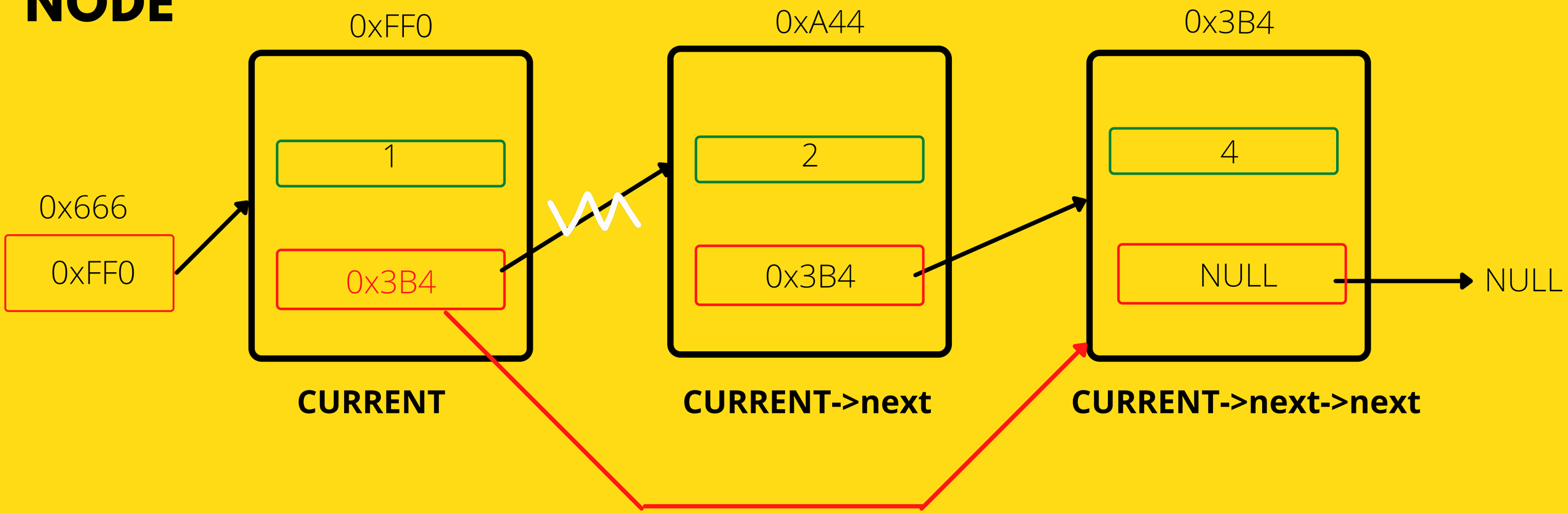
1. FIND THE NEXT NODE WITH DATA 2 TO DELETE



1st loop

LINKED LISTS (DELETE 2 - IN THE MIDDLE OF TWO NODES)

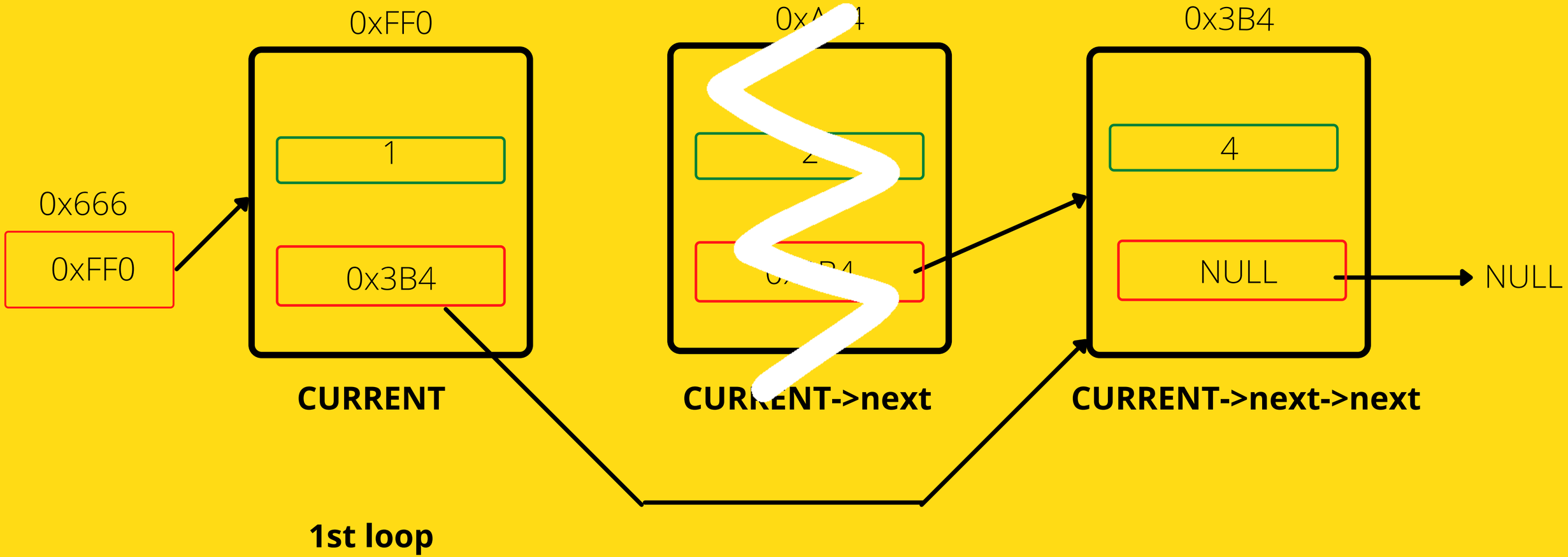
2. SET CURRENT NODE NEXT TO THE NEXT->NEXT NODE



1st loop

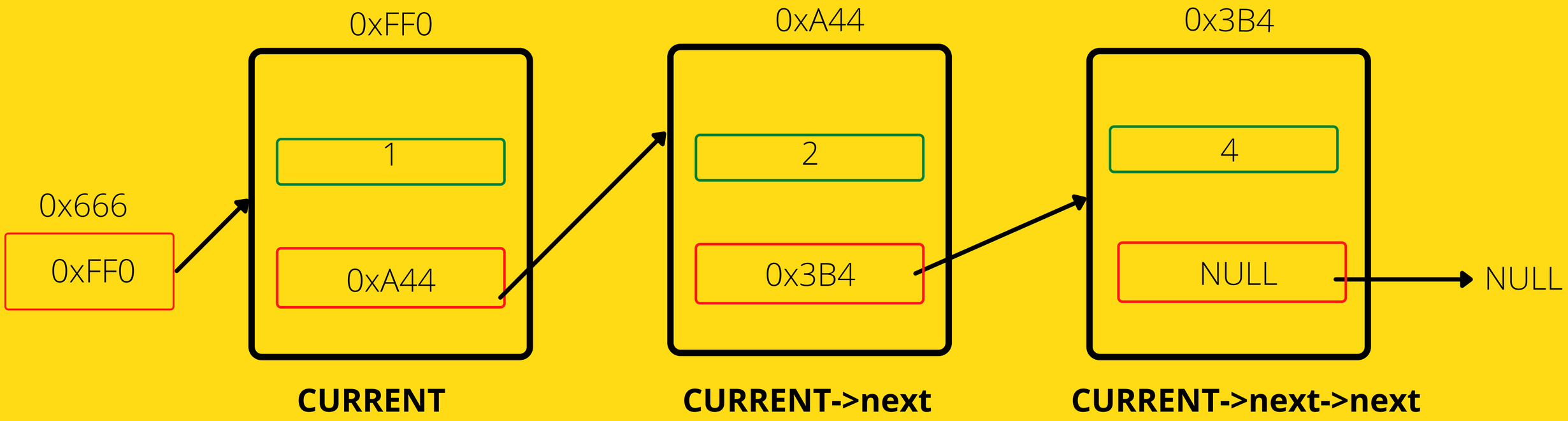
LINKED LISTS (DELETE 2 - IN THE MIDDLE OF TWO NODES)

3. FREE CURRENT->NEXT NODE



LINKED LISTS (DELETE 4 - THE TAIL)

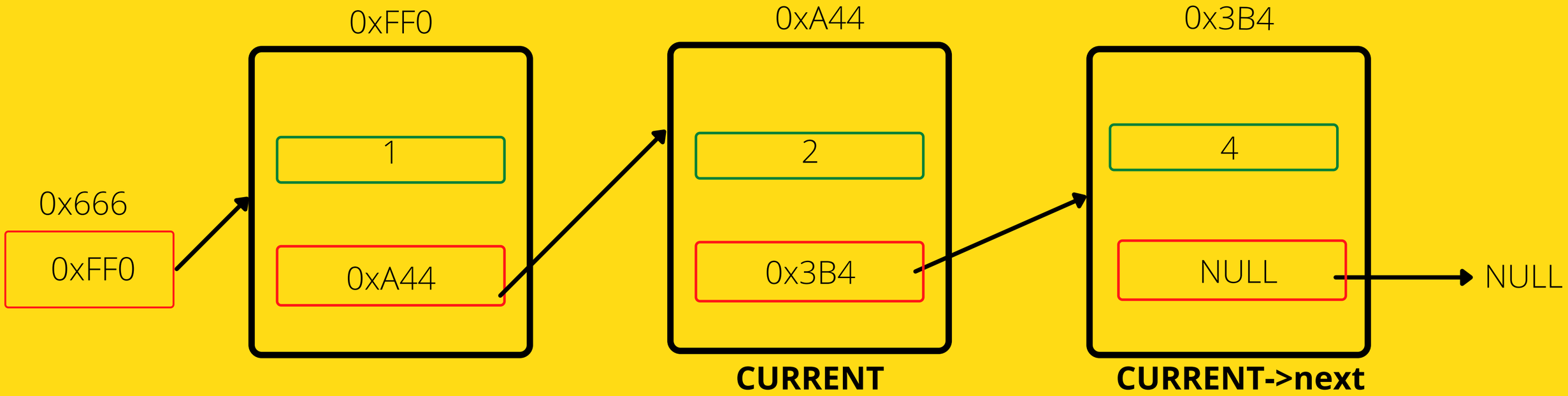
1. FIND THE NODE YOU WANT TO DELETE



1st loop

LINKED LISTS (DELETE 4 - THE TAIL)

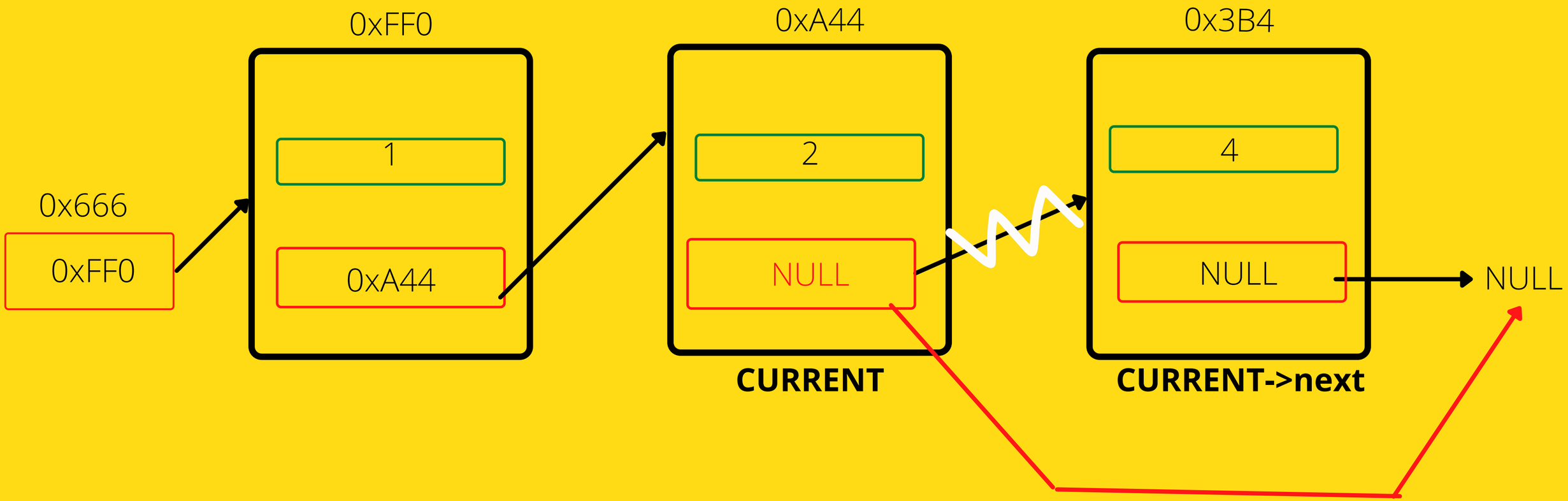
1. FIND THE NEXT NODE YOU WANT TO DELETE (IF IT'S NEXT IS NULL, IT IS THE TAIL)



2nd loop

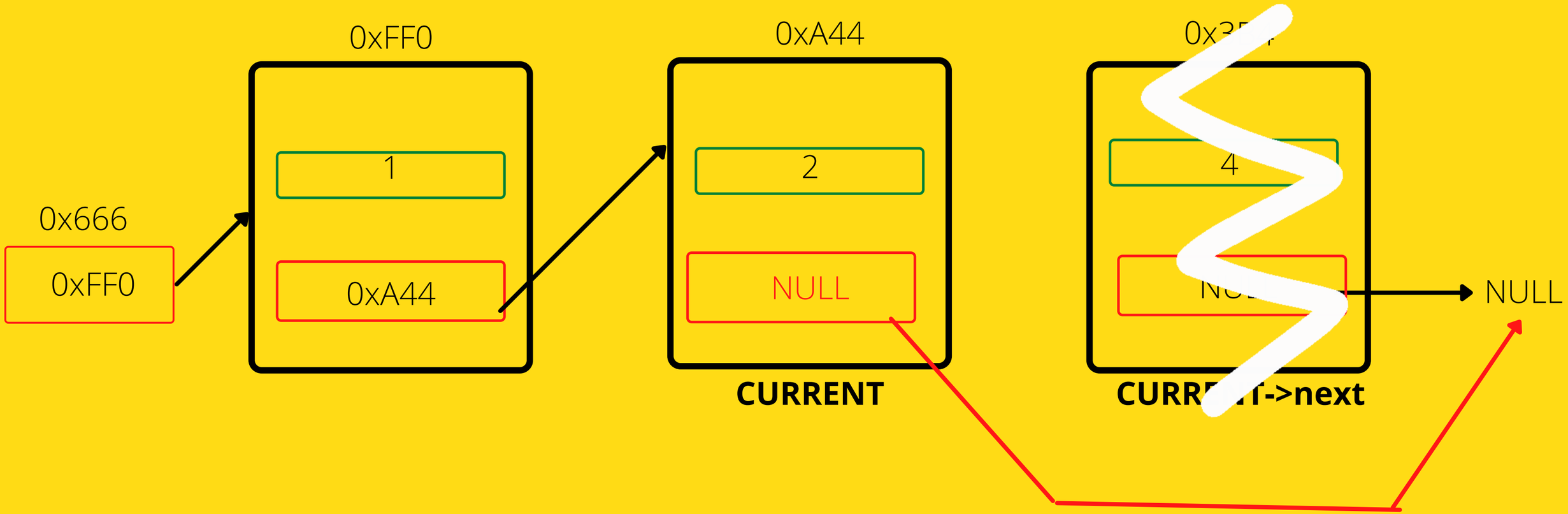
LINKED LISTS (DELETE 4 - THE TAIL)

2. POINT CURRENT NODE TO NULL



LINKED LISTS (DELETE 4 - THE TAIL)

3. FREE THE NODE



2nd loop

LINKED LISTS

DELETING A NODE IN SUMMARY

In all instances, we follow a similar structure of what to do when deleting a node. Please draw a diagram for yourself to really understand what you are deleting and the logic of deleting in a particular way.

To delete a node in a linked list:

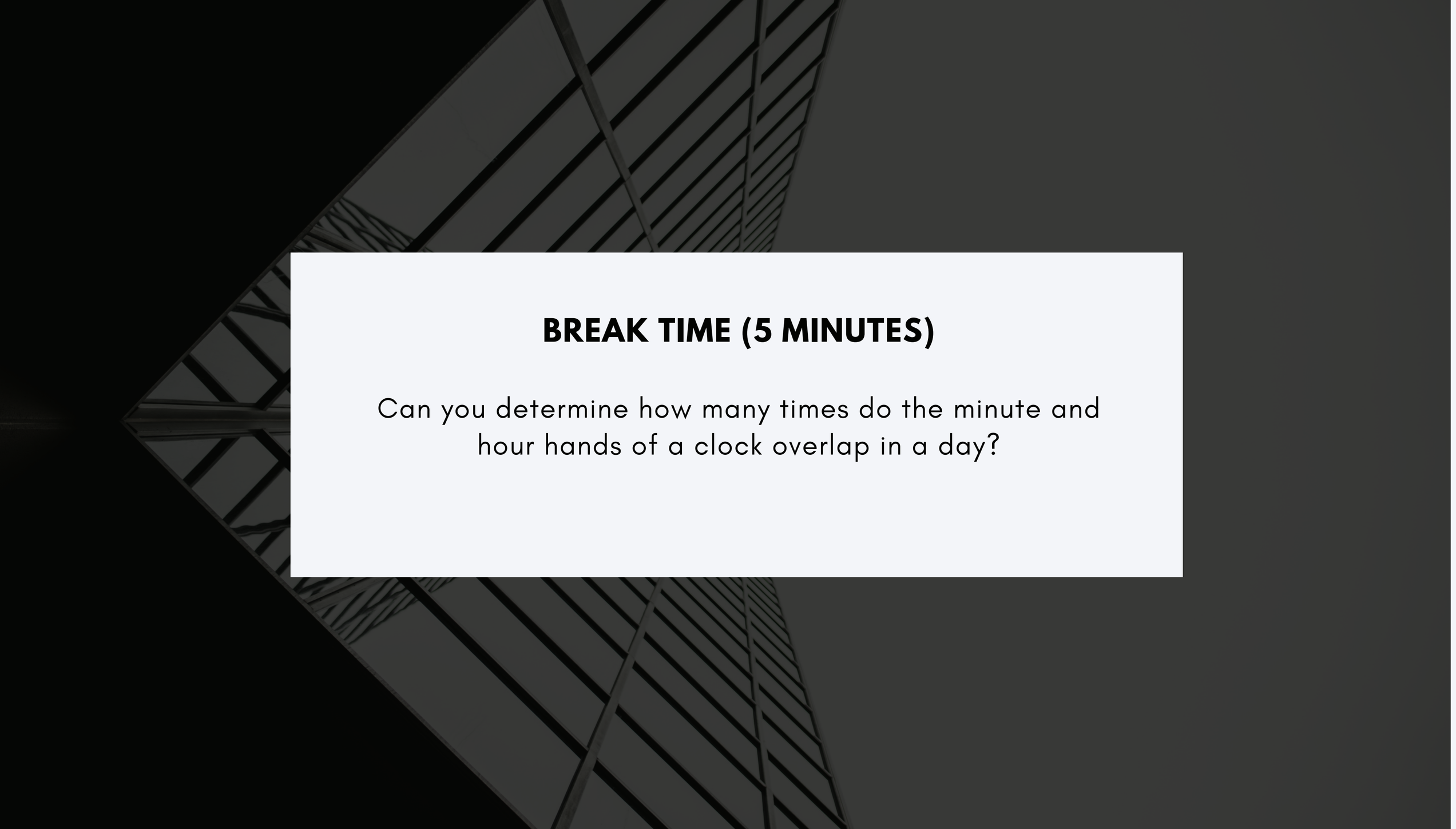
1. Find the previous node to the one that is being deleted
2. Change the next of the previous node
3. Free the node that is to be deleted

LINKED LISTS

LET'S SEE THE CODE

linked_list.c

```c
struct node *delete_node (struct node *head, int data) {
    //create a current pointer that is set to the head of the list
    struct node *current = head;
    // if there is nothing in the list
    if (current == NULL) {
        return NULL;
    // deleting at the head of the list
    } else if (current->data == data) {
        struct node *new_head = current->next;
        free(current);
        return new_head; //will return whatever was after current as the new head
    // if there is only one node in the list and it is the one to be deleted
    // above will capture it.
    }
    //otherwise start looping through the list to find the data
    //1. find the previous node to the one you want to delete
    while (current->next->data != data && current->next->next != NULL) {
        current = current->next;
    }
    //2. if the next node is the one to be deleted
    if (current->next->data == data) {
        // create a pointer to the new next
        struct node *new_next = current->next->next;
        // 3. free the node to be deleted
        free(current->next);
        //point the next node to the new pointer
        current->next = new_next;
    }
    return head;
}
```



BREAK TIME (5 MINUTES)

Can you determine how many times do the minute and hour hands of a clock overlap in a day?

PROBLEM TIME

Let's go back to our FIFA World Cup problem, and practice searching through the linked list for certain conditions and deleting new nodes in different places.

FEEDBACK?

PLEASE LET ME KNOW ANY FEEDBACK FROM TODAY'S LECTURE!

www.menti.com

Code: 26 63 47 7

WHAT DID WE LEARN TODAY?

LINKED LIST: SEARCHING FOR WHERE TO DELETE AND DELETING

linked_list.c

CONTINUING WITH THE HARDER PROBLEM

world_cup_prep.c

ANY QUESTIONS?

DON'T FORGET YOU CAN ALWAYS EMAIL US ON CS1511@CSE.UNSW.EDU.AU FOR ANY ADMIN QUESTIONS

PLEASE ASK IN THE FORUM FOR CONTENT RELATED QUESTIONS