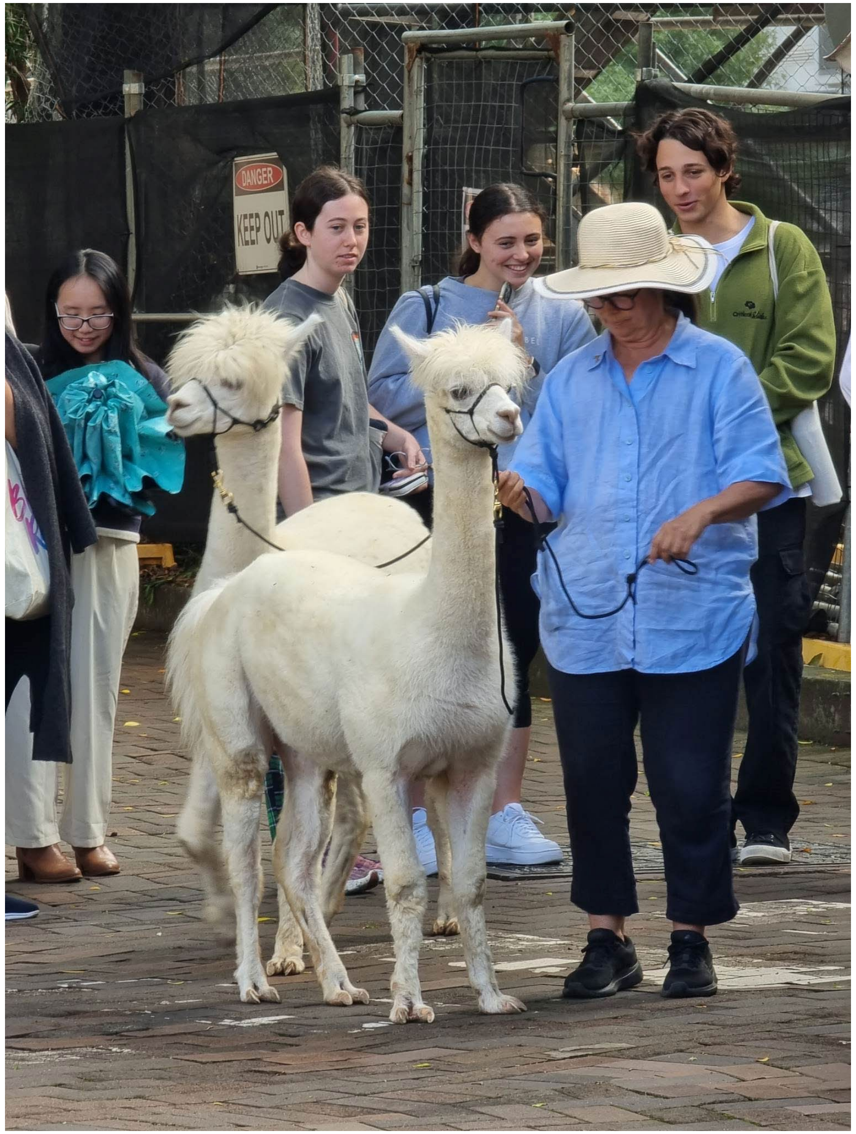
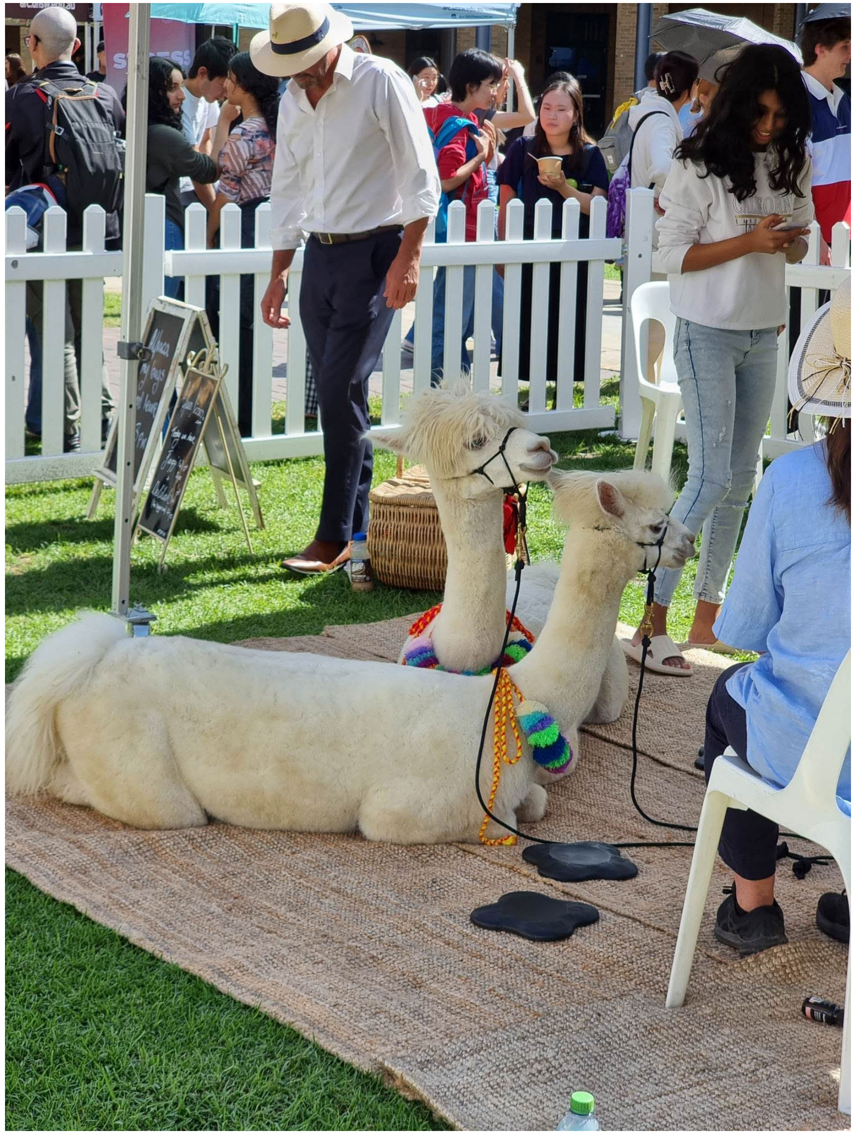
survey: which picture shows their size





COMP1511 Week 8!

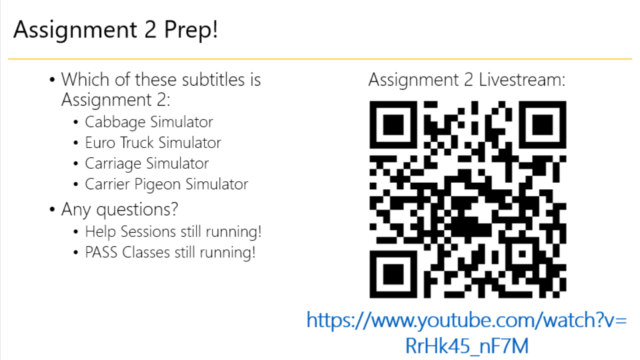
Stress Less Say Less

My GitHub:

https://github.com/william-o-s/unsw_comp1511_tutoring

The Agenda

Assignment 2



Assignment 2 Prep!

- Which of these subtitles is Assignment 2:
  - Cabbage Simulator
  - Euro Truck Simulator
  - Carriage Simulator
  - Carrier Pigeon Simulator
- Any questions?
  - Help Sessions still running!
  - PASS Classes still running!

Assignment 2 Livestream:

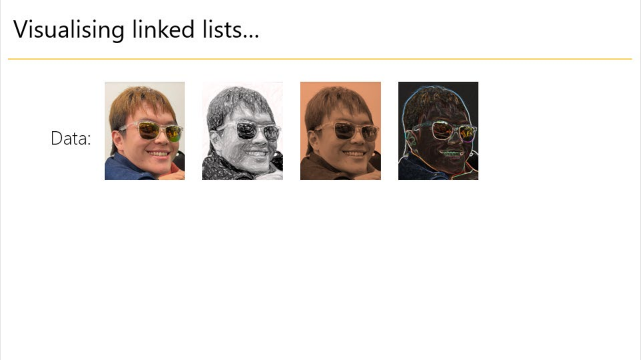
https://www.youtube.com/watch?v=RrHk45_nF7M

`malloc`

Recap: how does **`malloc`** work?

- Explain these lines:
  - `int *my_ptr = malloc(sizeof(int));`
  - `int *my_ptr = malloc(4);`
  - `int *my_ptr = malloc(sizeof(*my_ptr));`
  - `*my_ptr = 100;`
- Explain how to **`malloc`** these variables with the person next to you:
  - `char`
  - `int`
  - `int` array of size 5

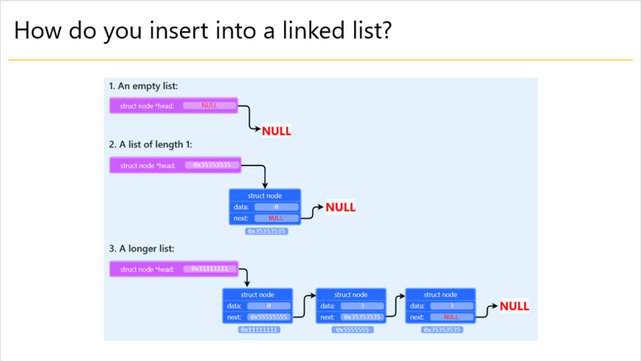
Drawing Linked Lists



Visualising linked lists...

Data:

Insertion into Linked Lists



How do you insert into a linked list?

Assignment 2 Prep!

- Which of these subtitles is Assignment 2:
  - Cabbage Simulator
  - Euro Truck Simulator
  - Carriage Simulator
  - Carrier Pigeon Simulator
- Any questions?
  - Help Sessions still running!
  - PASS Classes still running!

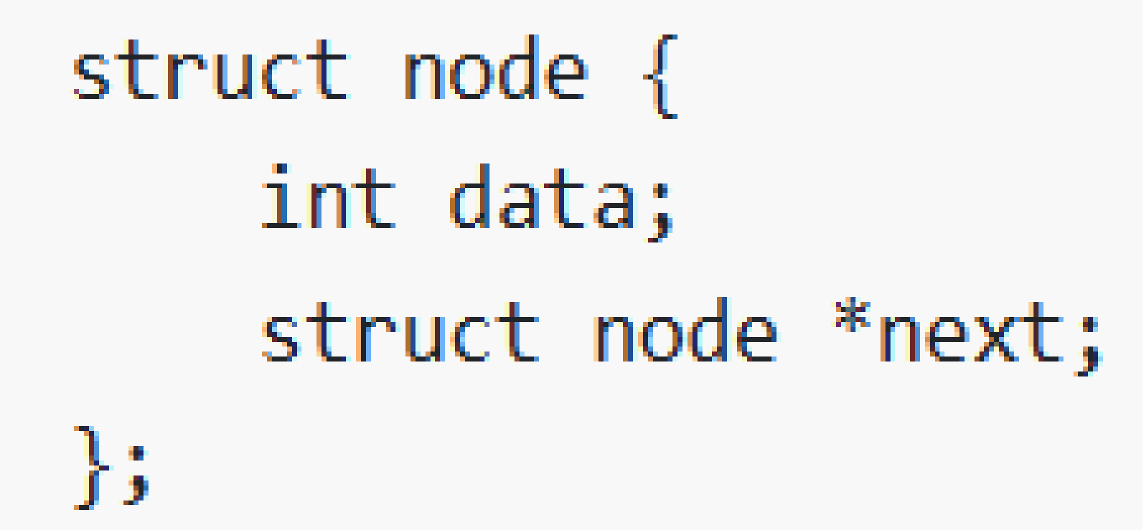
Assignment 2 Livestream:

https://www.youtube.com/watch?v=RrHk45_nF7M

Recap: how does **`malloc`** work?

- Explain these lines:
  - `int *my_ptr = malloc(sizeof(int));`
  - `int *my_ptr = malloc(4);`
  - `int *my_ptr = malloc(sizeof(*my_ptr));`
  - `*my_ptr = 100;`
- Explain how to **`malloc`** these variables with the person next to you:
  - `char`
  - `int`
  - `int` array of size 5

Recap: **`malloc`** for **`struct`**?



```
struct node {
    int data;
    struct node *next;
};
```

With the person next to you:

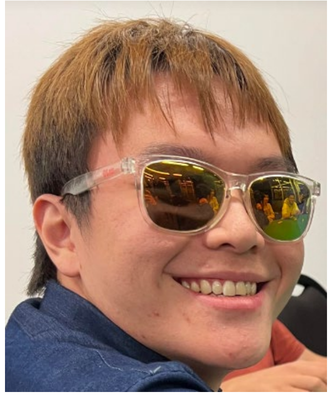
1. Write code to **`malloc`** this **`struct`**
2. Initialise the created **`struct`** with default values
3. Move your code into a function with prototype:

```
struct node *create_node(int data);
```

Why would we move this code into a function?

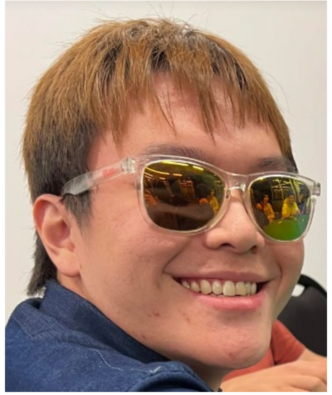
Visualising linked lists…

Data:

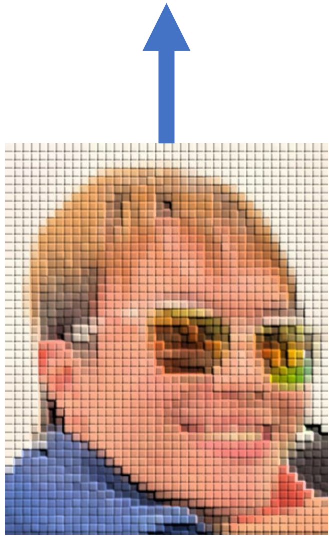


Visualising linked lists...

Data:

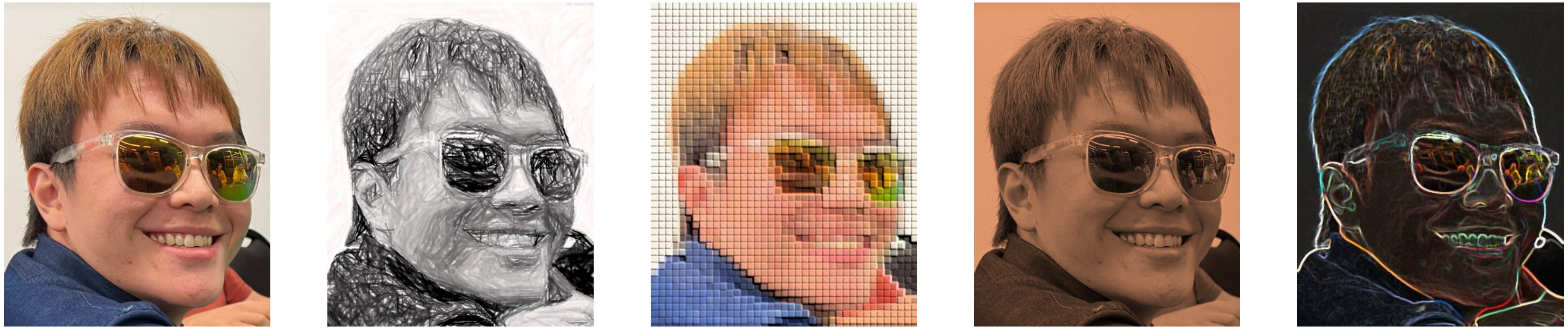


New data:



What type of collection is this?

Data:

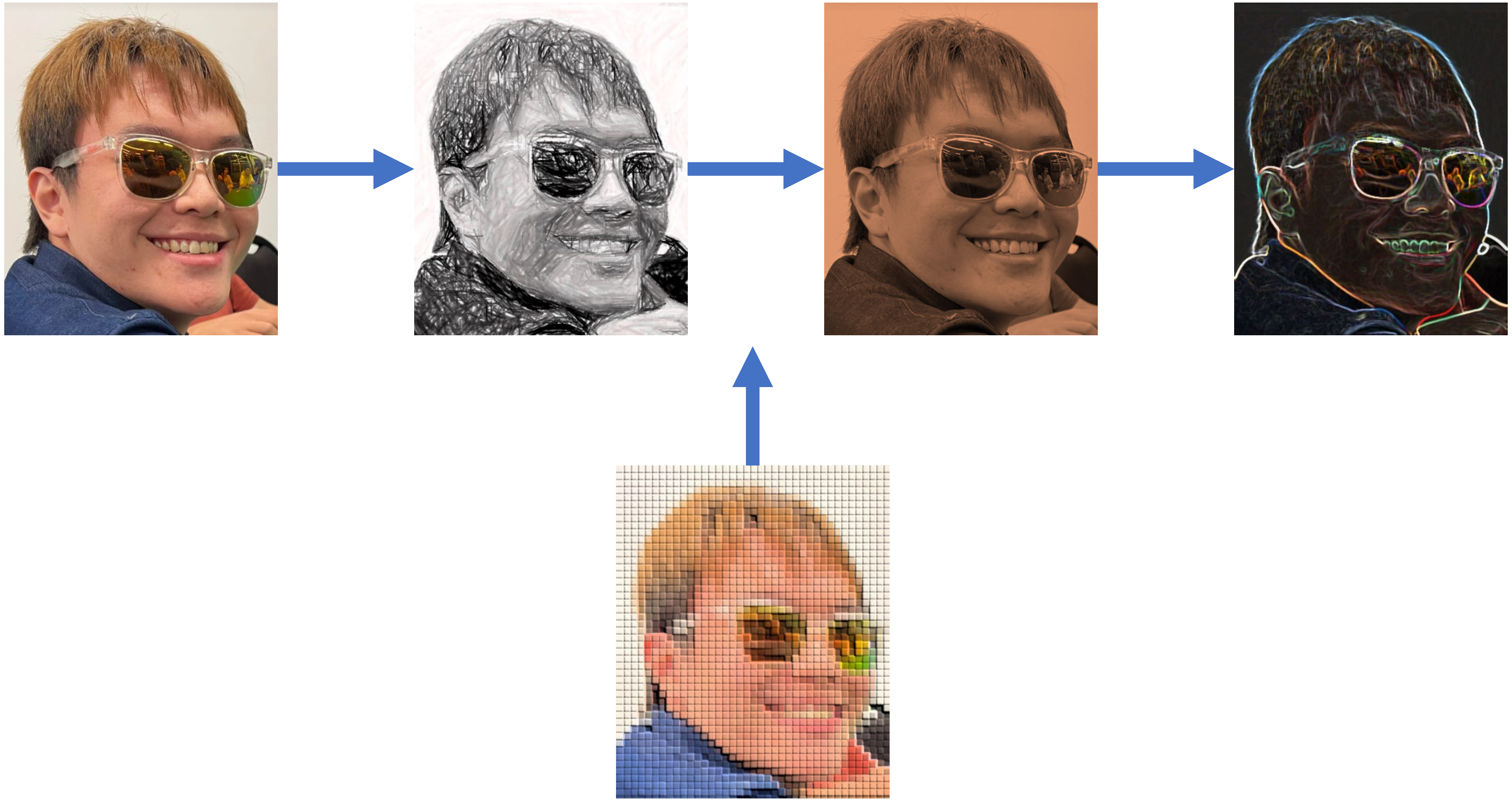


New data:

How would you know where to insert?

Data:

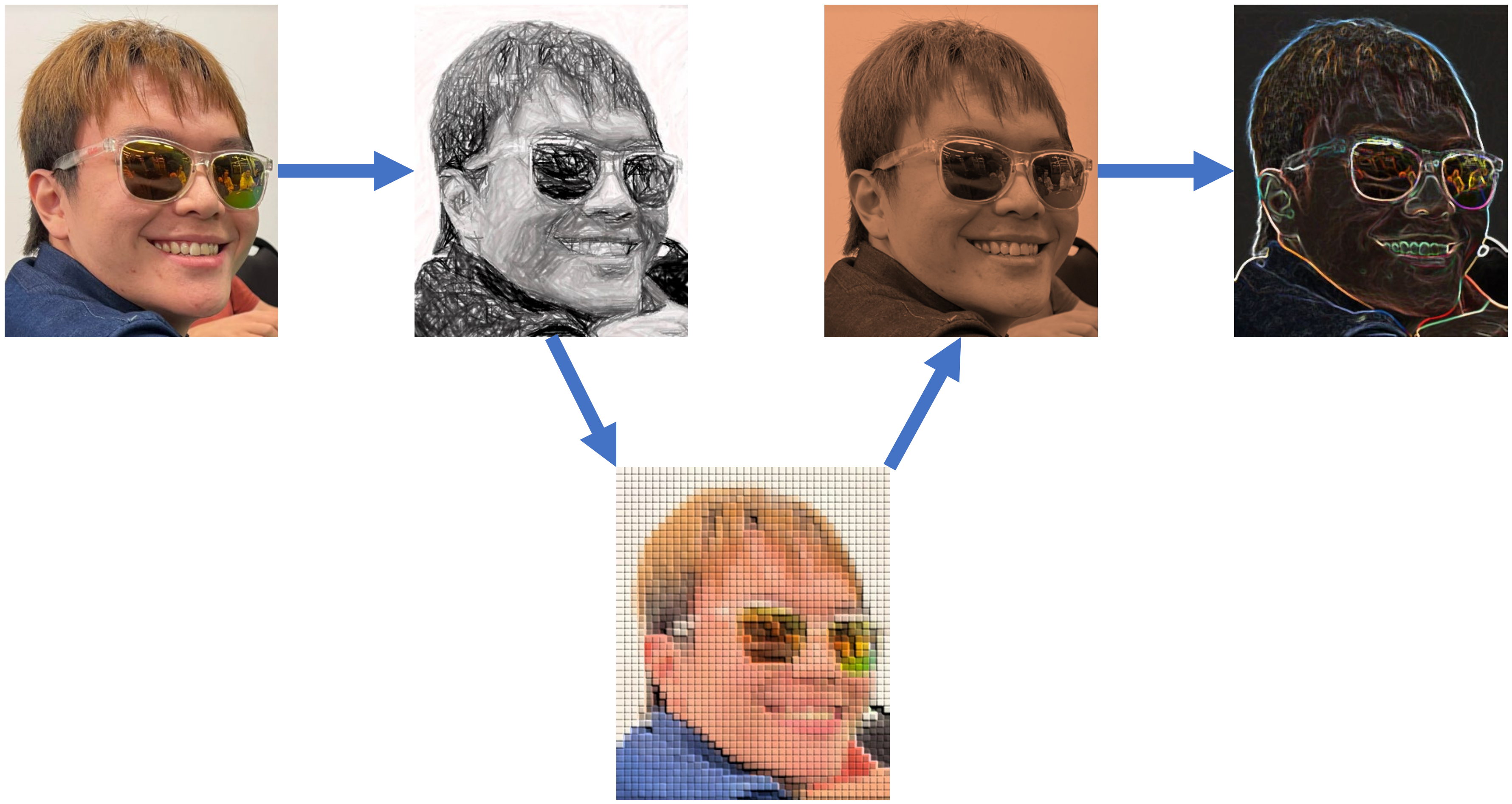
New data:



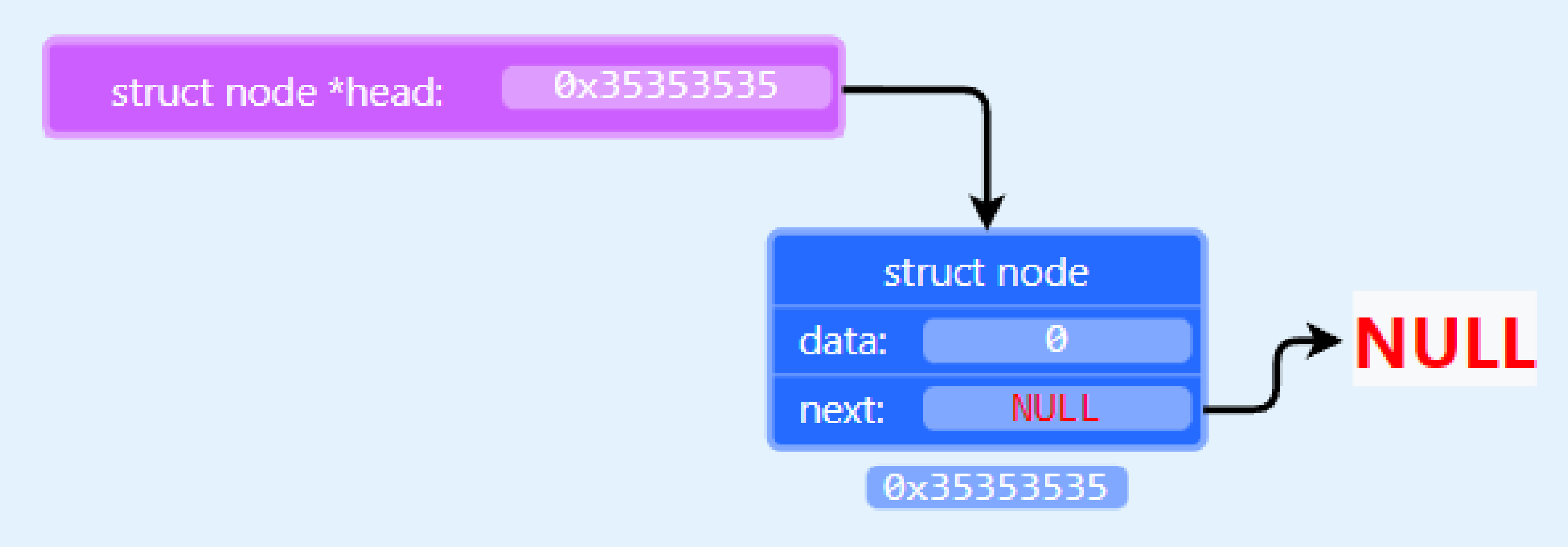
Which is the head node, the last node, the middle node?

Data:

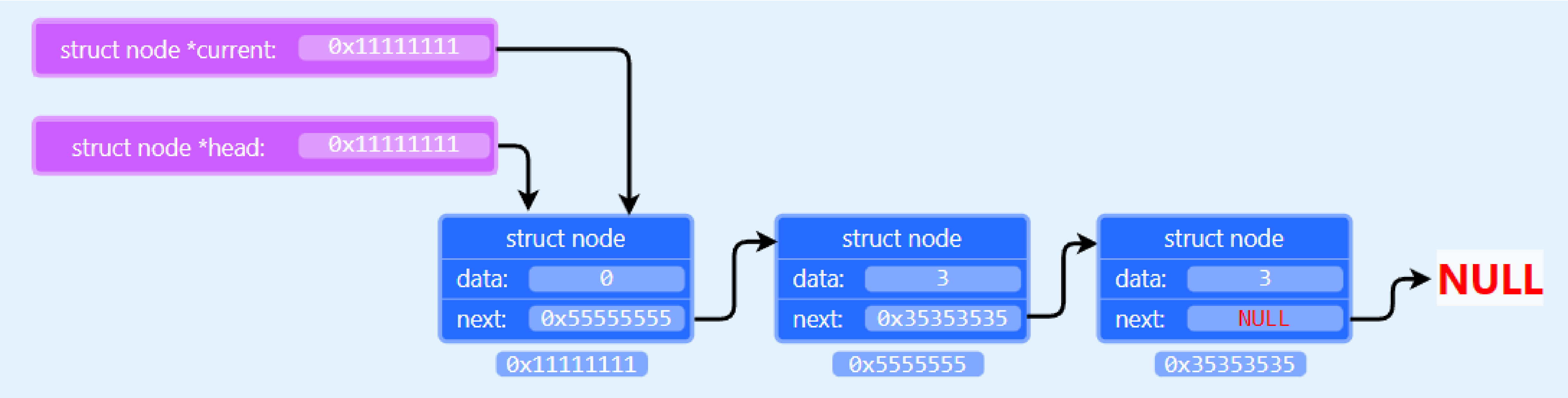
New data:



How would you modify the head node?



Why do we need a **`current`** node?



struct node *current: 0x11111111

struct node *head: 0x11111111

struct node
data: 0
next: 0x55555555
0x11111111

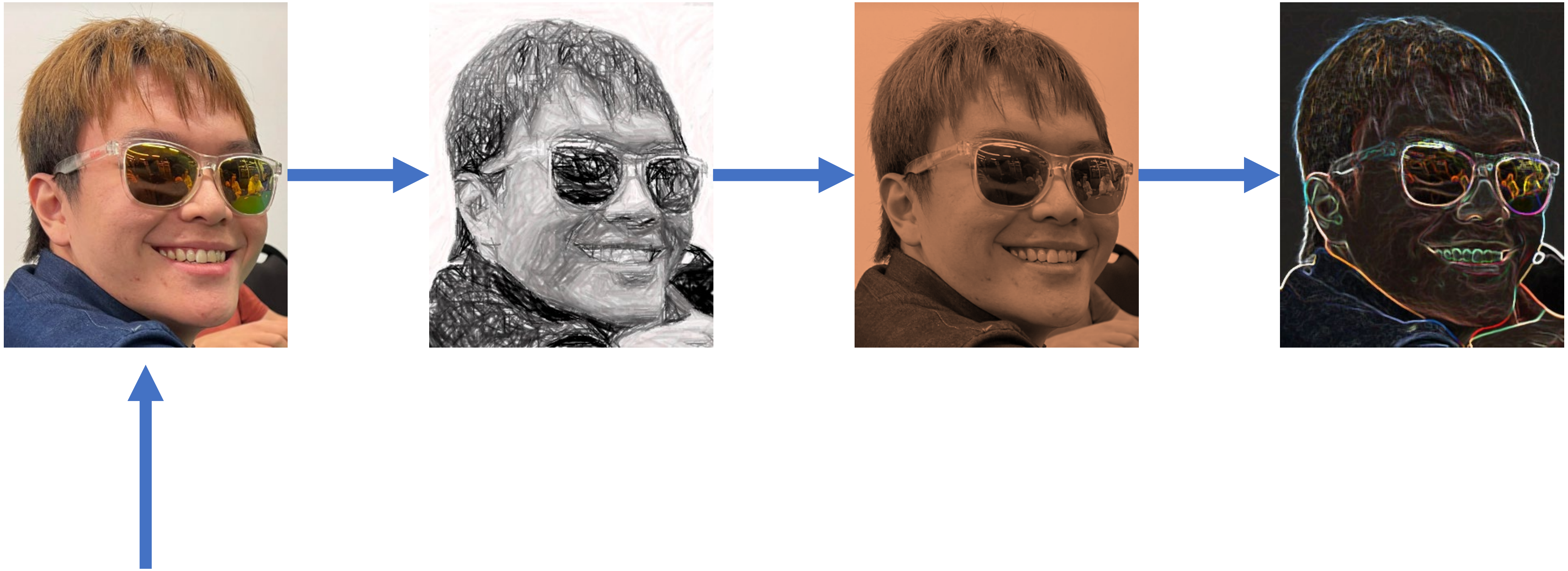
struct node
data: 3
next: 0x35353535
0x5555555

struct node
data: 3
next: NULL
0x35353535

NULL

How do we loop over a linked list?

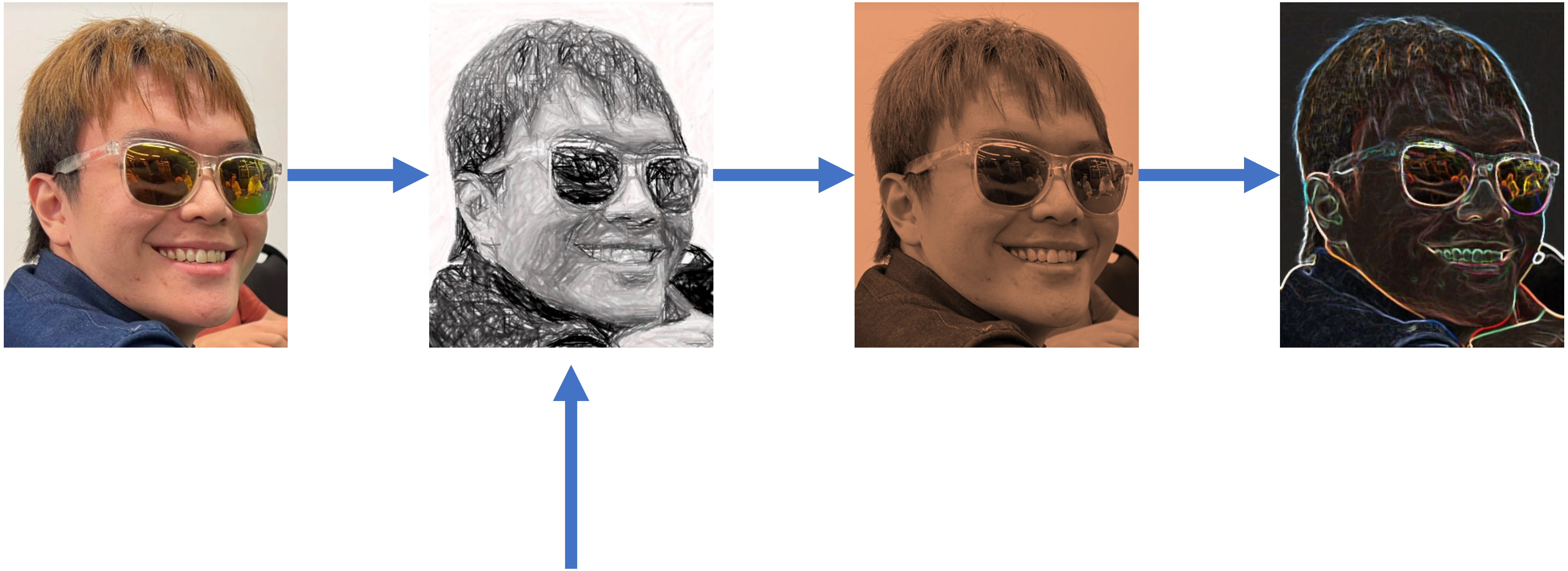
List:



`current`

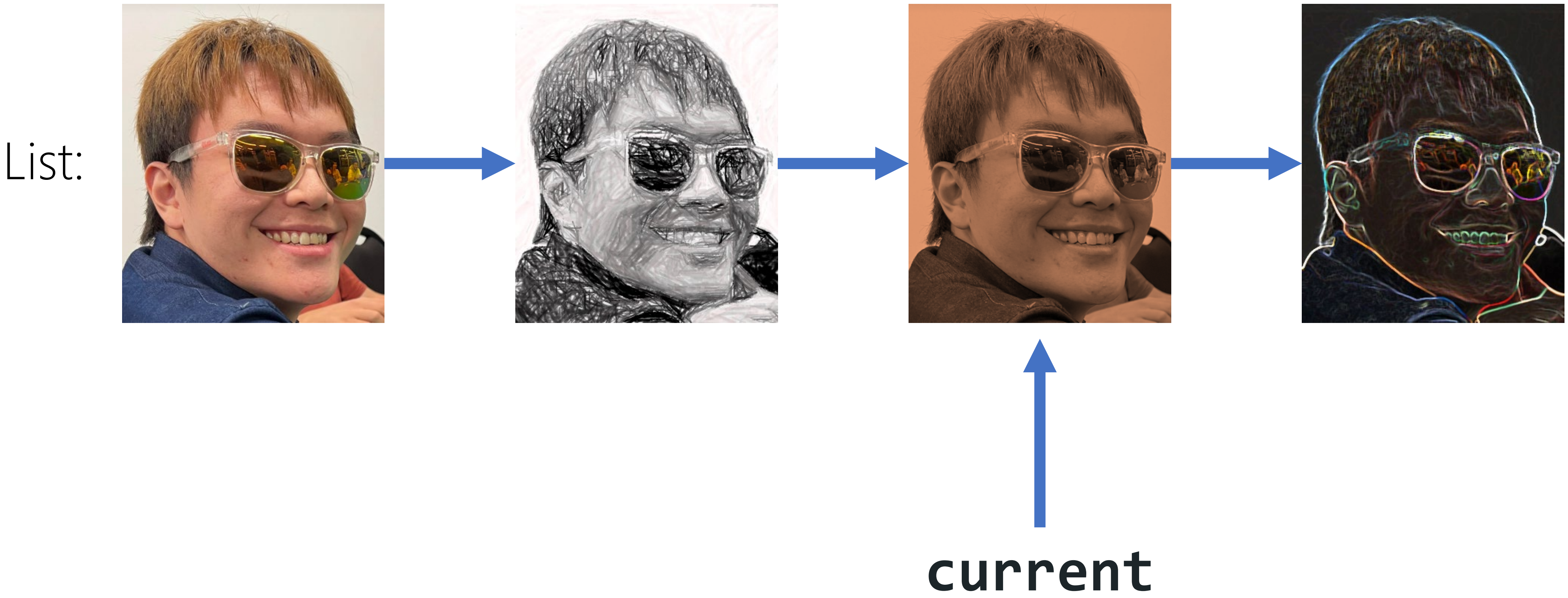
How do we loop over a linked list?

List:



`current`

How do we loop over a linked list?

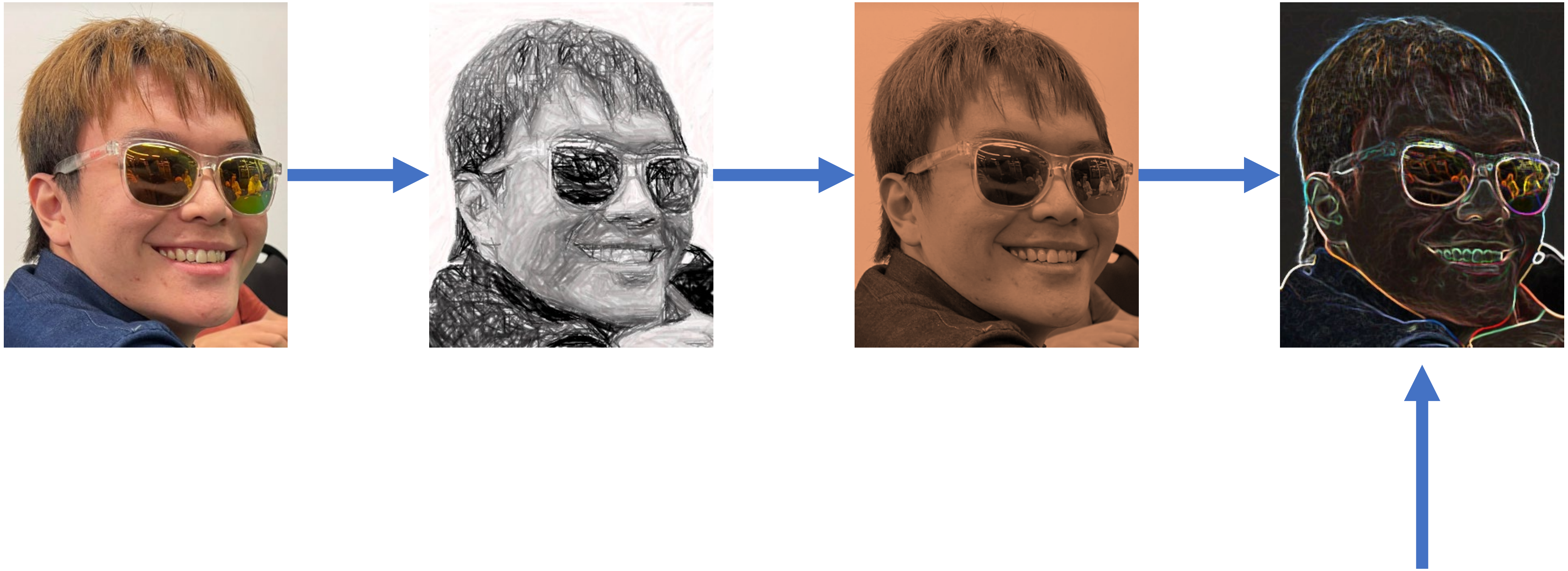


List:

`current`

How do we loop over a linked list?

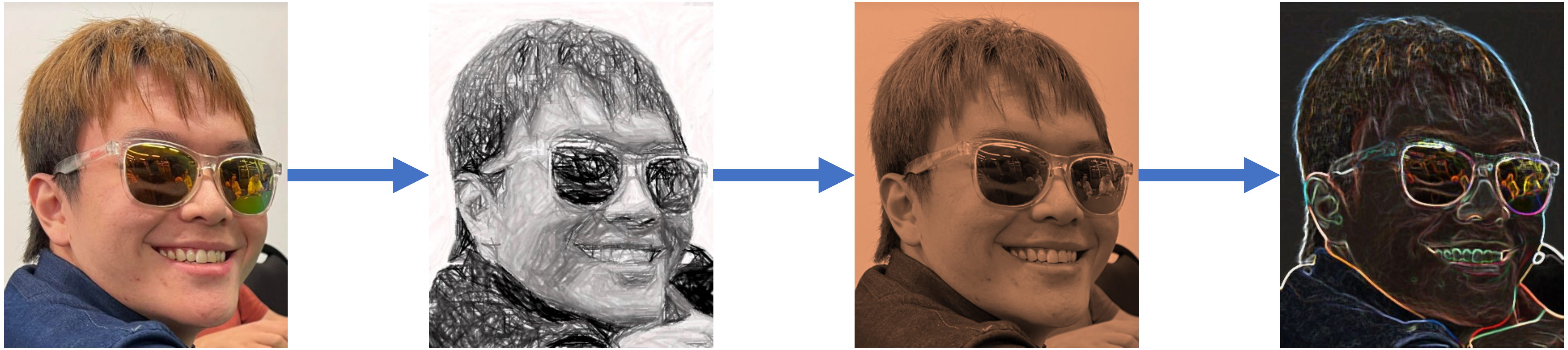
List:



`current`

How do we loop over a linked list?

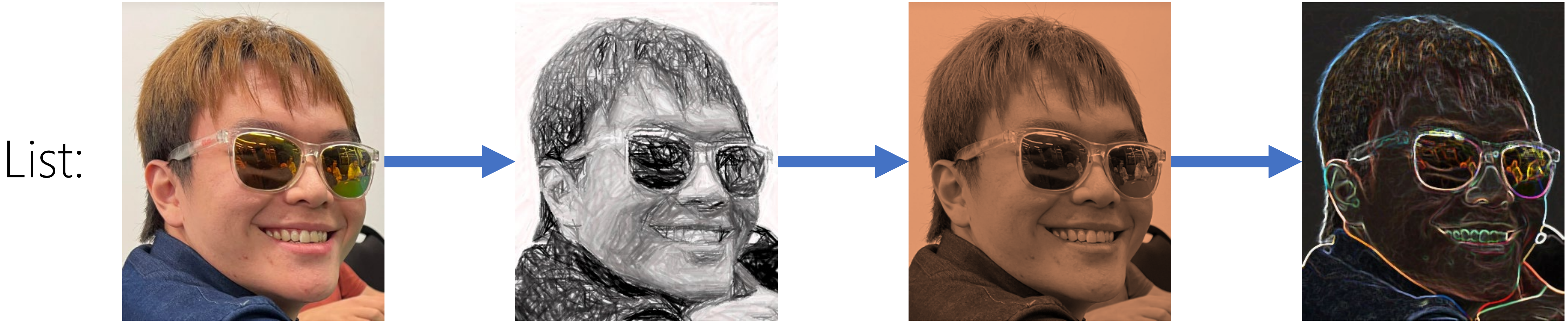
List:



Pseudocode:

```
set current to head
while current is not NULL
    move current to next node
```

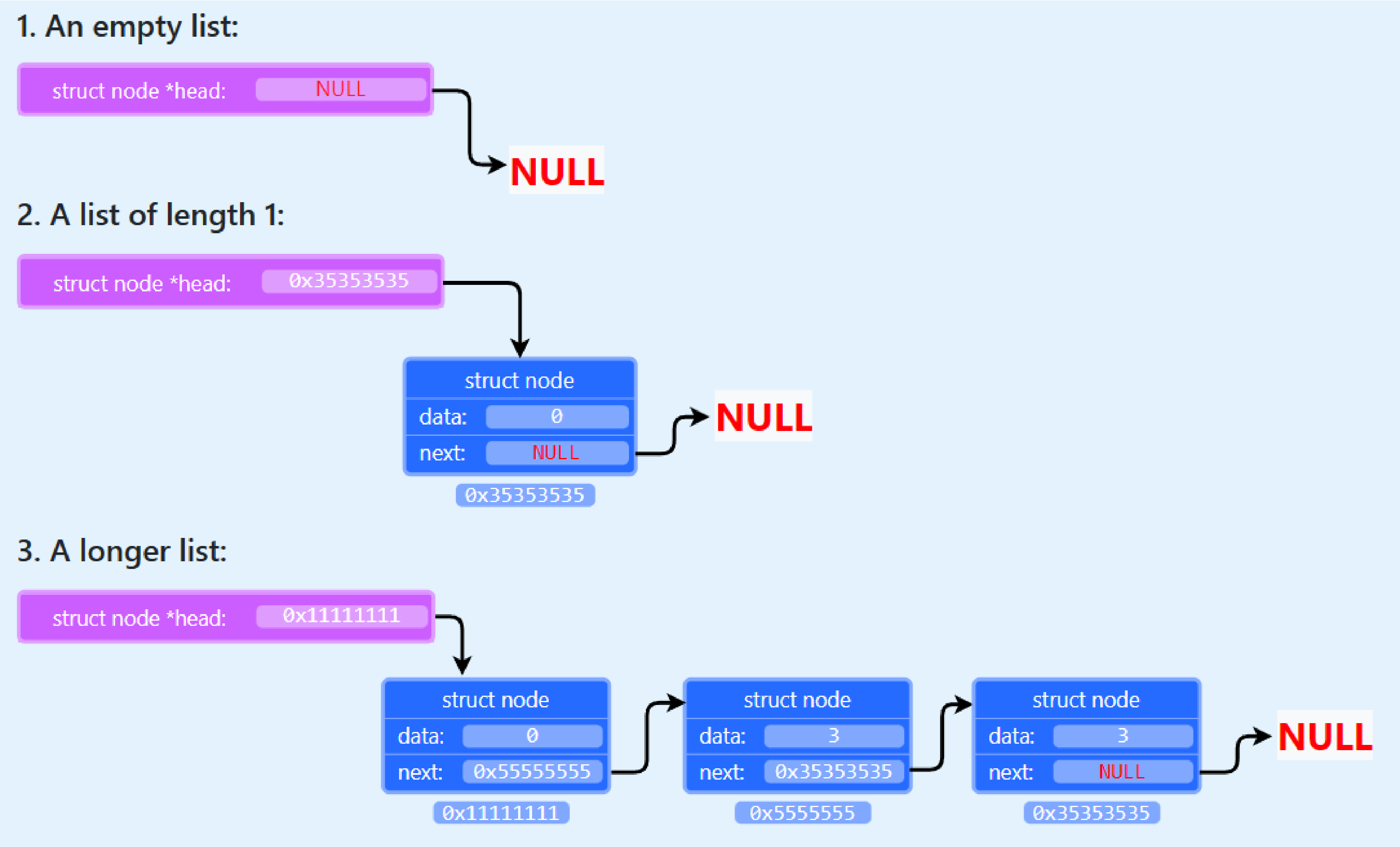
What if we want to stop at the last node?



```
Pseudocode (incorrect):
set current to head
While current next node is not NULL
    move current to next node
```

current

How do you insert into a linked list?

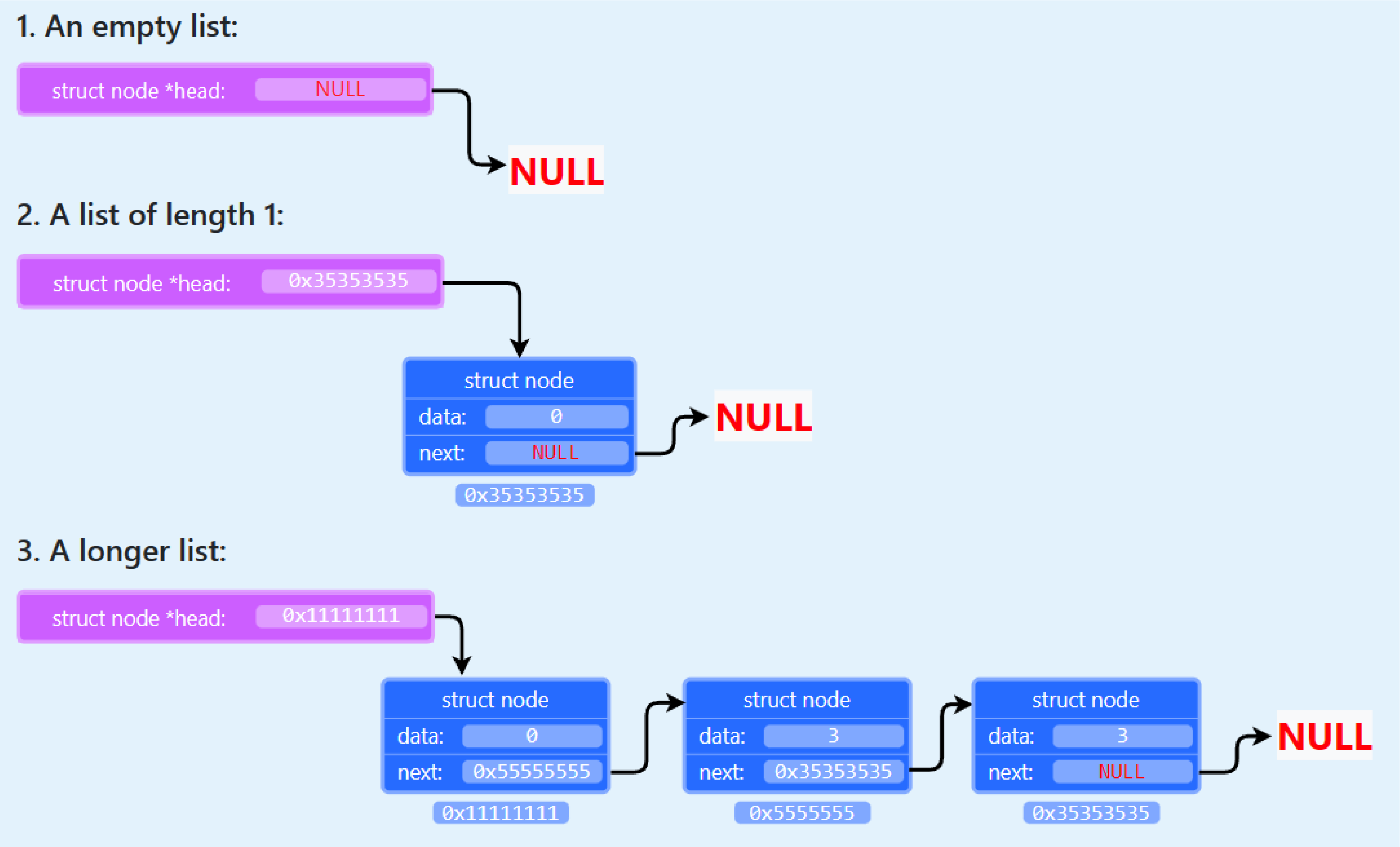


1. An empty list:

2. A list of length 1:

3. A longer list:

How do you insert into a linked list at index **n**?



- Edge cases:
  - n == 0
  - n > size of list
  - n >= 0 && empty list
  - N < size of list