i waited forever for this

COMP1511 Week 8!

H13A: 1pm – 4pm

Tutors: Me + Vivian Zheng

My GitHub:

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

Course Homepage:

https://cgi.cse.unsw.edu.au/~cs1511/23T3/

Time to start Assignment 2!

Source: The Crag

The Agenda

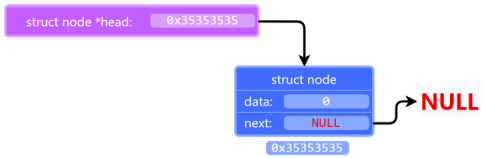
Malloc (10 mins)

Let's go over some memory concepts

- How much memory is needed for an `int` vs `double`?
  - What function/keyword do we use to check this?
- How much memory is needed for 10 integers?
  - Can you make this into a formula?
- How do you get C to give you this memory?
  - What function do we use to do this?
  - What parameters are required? What does it return?

Diagramming LLs (20 mins)

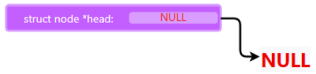
What are some interesting things in this diagram?



LL Insertion (15 mins)

When inserting new nodes, you could run into this scenario...

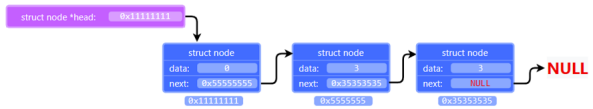
1. An empty list:



LL Practice (15 mins)

Can you write the pseudocode for this case?

3. A longer list:



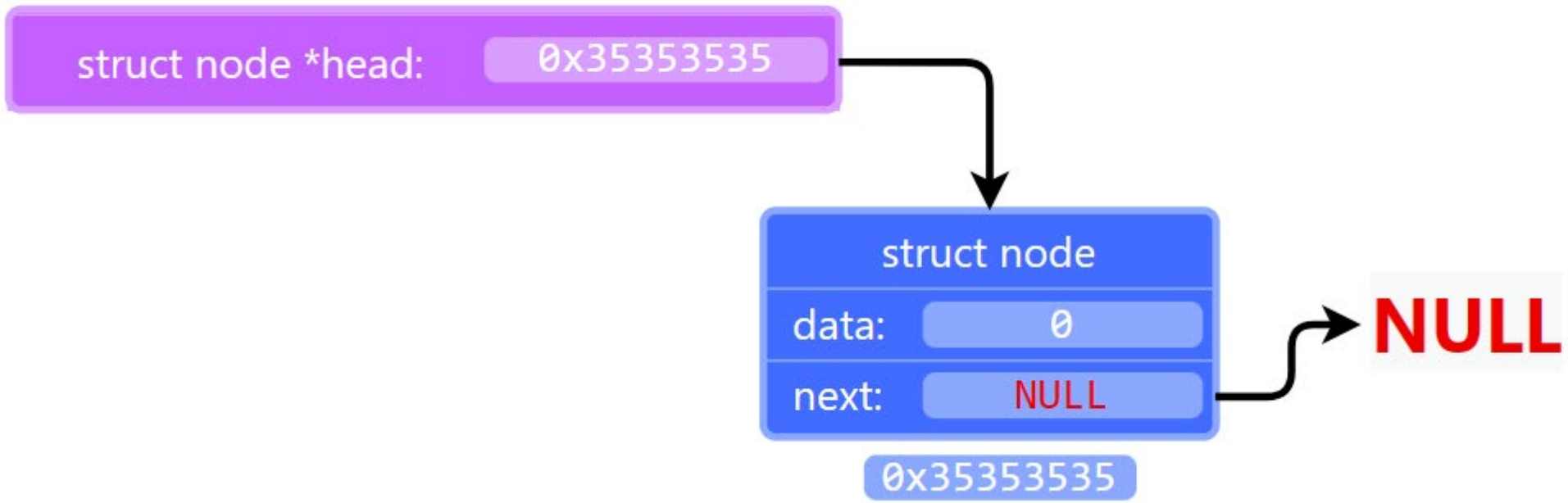
Let’s go over some memory concepts

- How much memory is needed for an **`int`** vs **`double`**?
  - What function/keyword do we use to check this?
- How much memory is needed for 10 integers?
  - Can you make this into a formula?
- How do you get C to give you this memory?
  - What function do we use to do this?
  - What parameters are required? What does it return?

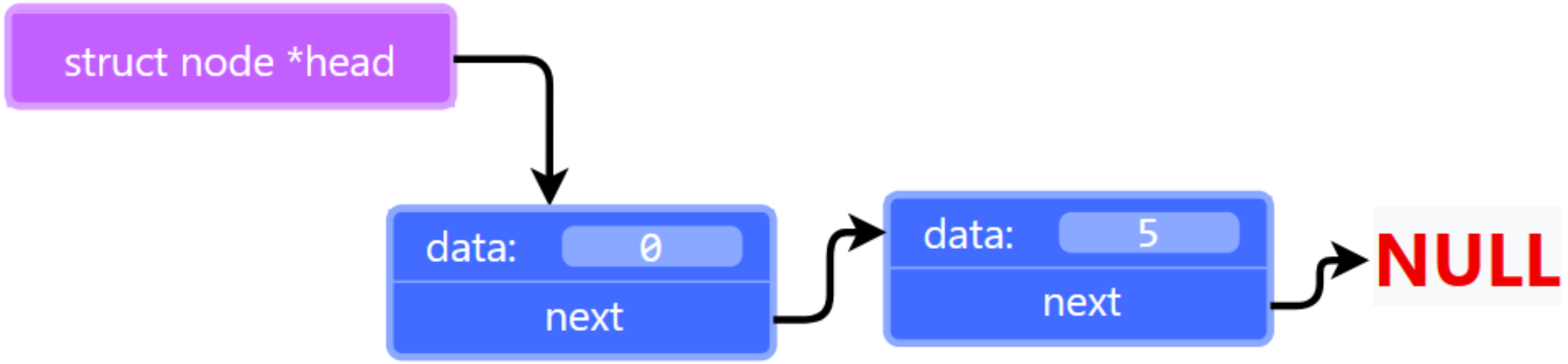
Note: look at this after malloc.c

- When to use **`malloc`**?
  - Dynamic array size
  - Can change size of array (not covered in 1511)
  - Can return a pointer
  - Can use helper functions with linked lists (coming soon)
- When not to use **`malloc`**?
  - Simple
  - Memory leaks
- Let's create a **`struct node`** now!

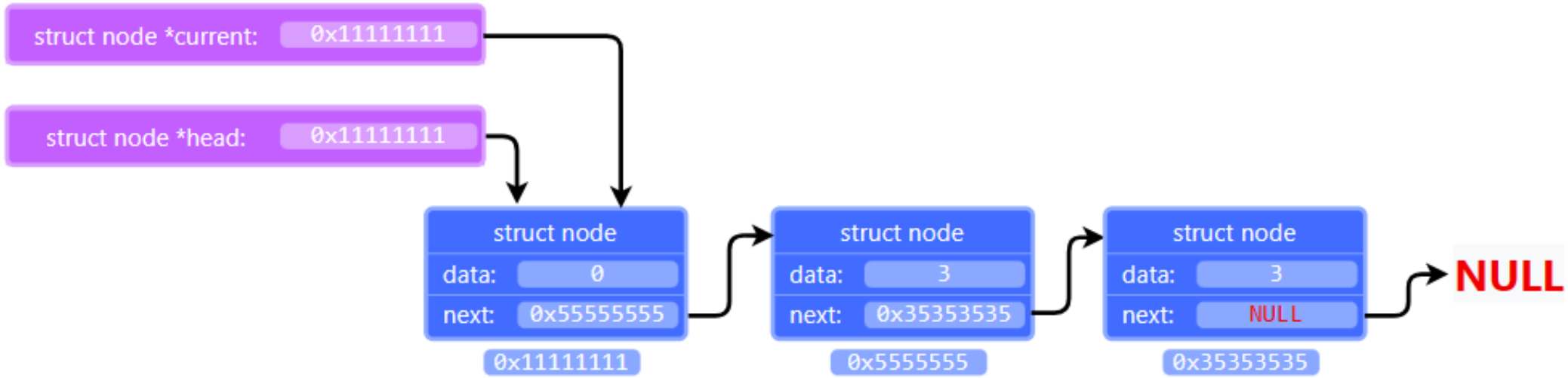
What are some interesting things in this diagram?



How would insertion work visually? How do we get here?



What about iterating? What's special about the tail node?

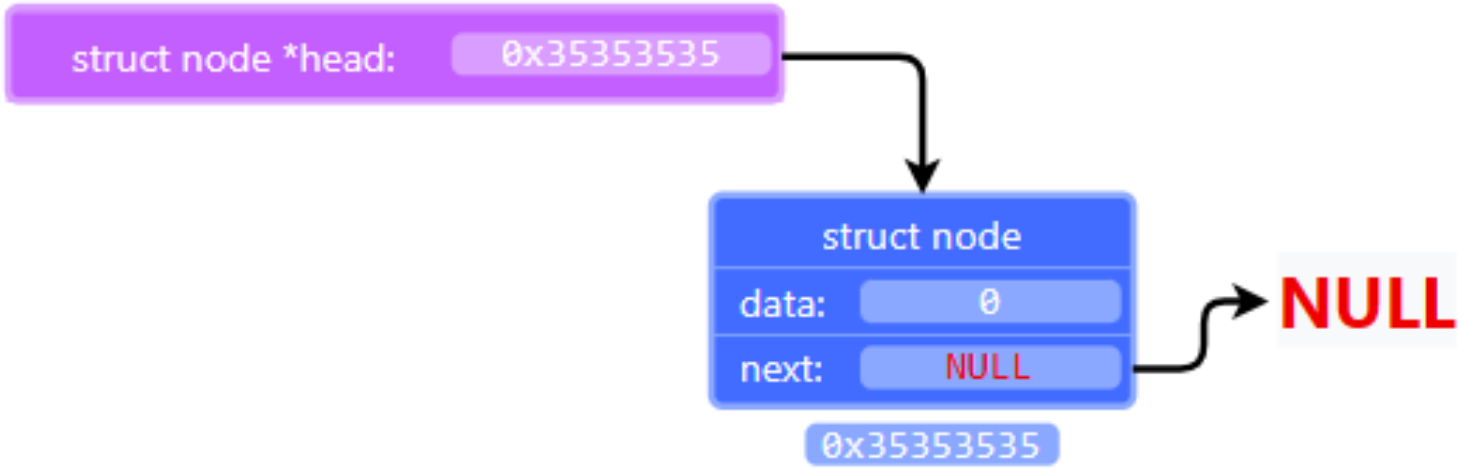


When inserting new nodes, you could run into this scenario…

1. An empty list:

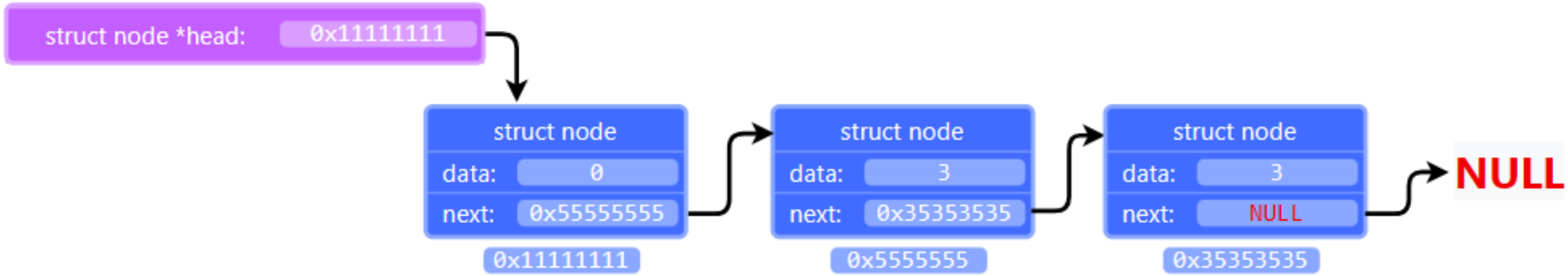
...or this scenario...

2. A list of length 1:



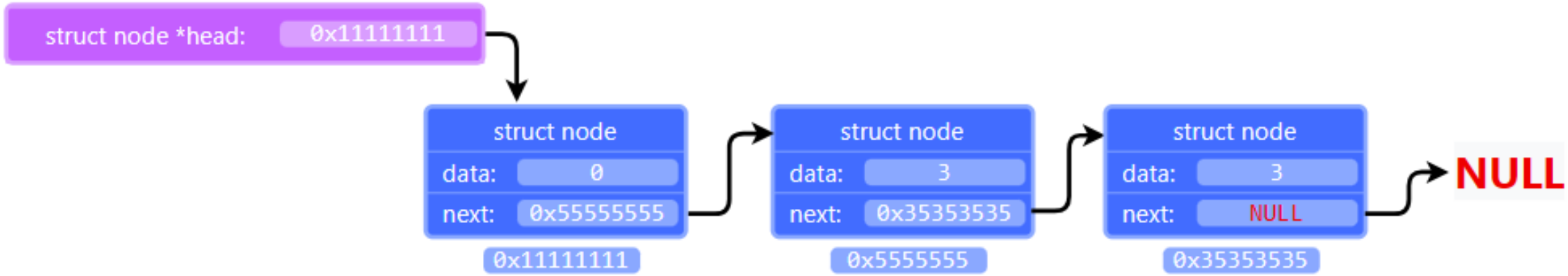
...or this last third scenario

3. A longer list:



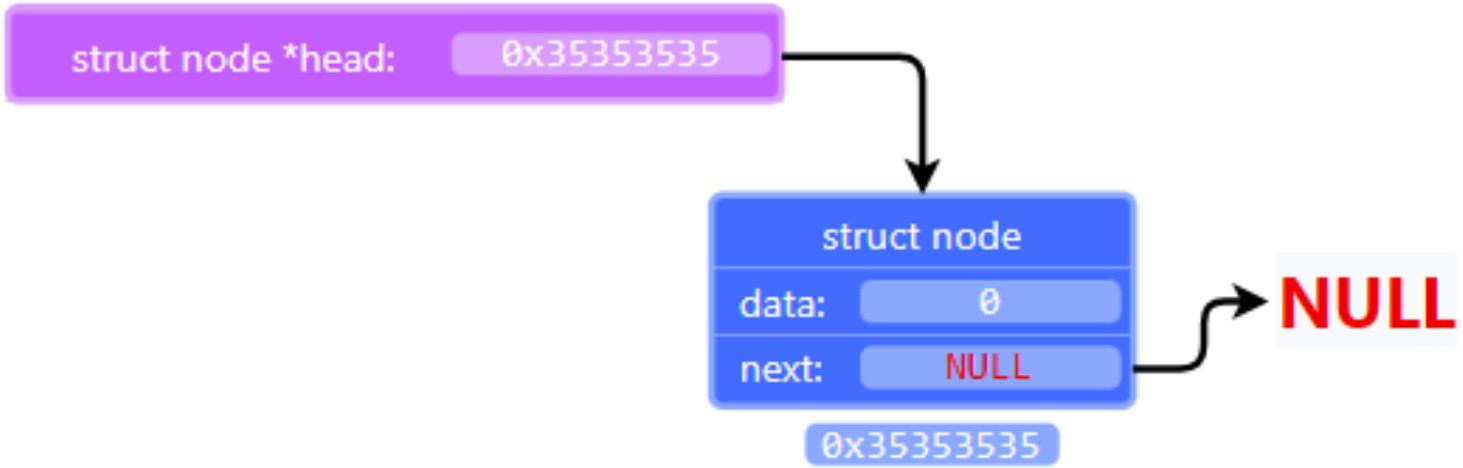
Can you write the pseudocode for this case?

3. A longer list:



Do you need to modify your pseudocode for this case?

2. A list of length 1:



What about for this case?

1. An empty list:

VSCode Shortcuts

- Start with Ctrl+Shift+P
  - "Toggle Multi-Cursor Editor"
  - Convert text casing: (highlight text) → Ctrl + Shift + P → "Transform to …"
- Multiple Cursors: Ctrl + Click anywhere
  - Cursor over multiple lines vertically: Shift + Alt + Click on line
- Duplicate Line: Ctrl + Shift + Alt + Up/Down Arrow
- Move Lines: Alt + Up/Down Arrow
- Change All Occurrences: Ctrl + Shift + L or Ctrl + D
- Indentation: (Highlight line/lines) → Ctrl + Left/Right Square Bracket
- Find and Replace: Ctrl + F → (click dropdown) → Replace next