Data Structures

Linked Lists

CS 225

August 30, 2023

Brad Solomon & G Carl Evans

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Department of Computer Science

Exam 0 (August 29 — 31)

An introduction to CBTF exam environment / expectations

Quiz on foundational knowledge from all pre-reqs

Practice questions can be found on PL

Topics covered can be found on website

If you haven't yet signed up do so ASAP!

Exam 1 (September 11 — 13)

A mixture of multiple choice** and coding questions

Exam on content up to **September 4th**

Prairielearn will have a practice exam sometime next week

Sign up also began August 24th

MP_stickers (Due September 11th)

An introductory assignment

Consider the Rule of Three (Rule of Zero)

Good practice on defining classes

A lot of the functions here are simple — don’t share code!

Make sure you understand **PNG** and **HSLAPixel**

Be Respectful: Noise Levels

Class runs from 11:00 — 11:50 AM

The last minutes of lecture normally wraps up a point, opens the floor for questions, or asks you to think critically about a topic we will start the following class. **All of this is important!**

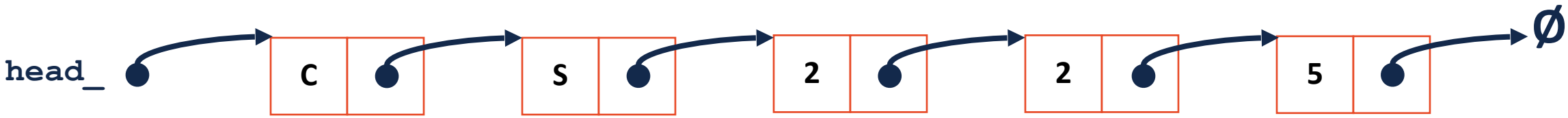
Please don’t start to leave until class has wrapped up for the day

Learning Objectives

Review linked list operations (and go over new ones)

Introduce array list implementations

Linked List Review

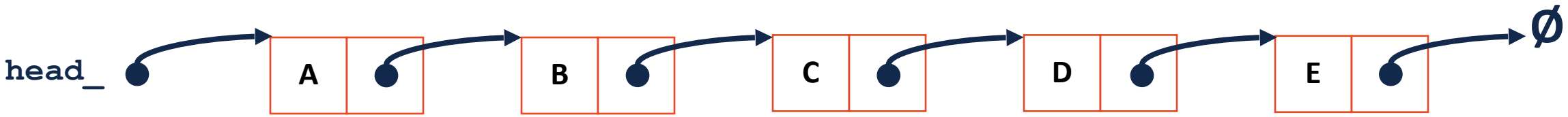


1. The Linked List is **singly linked**

2. The Linked List does not permit **random access**

3. _index(index) returns a **reference to a pointer**

Linked List: `insert(data, index)`

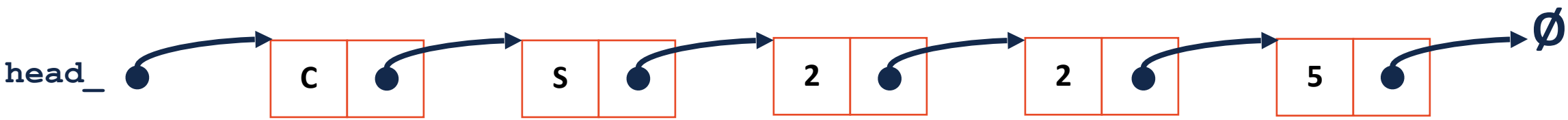


List Random Access [ ]

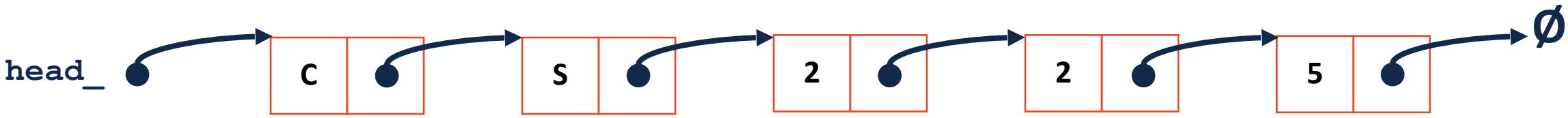
Given a list L, what operations can we do on L[ ]?

List.hpp

```
48 template <typename T>
49 T & List<T>::operator[](unsigned index) {
50
51
52
53
54
55
56
57
58 }
```

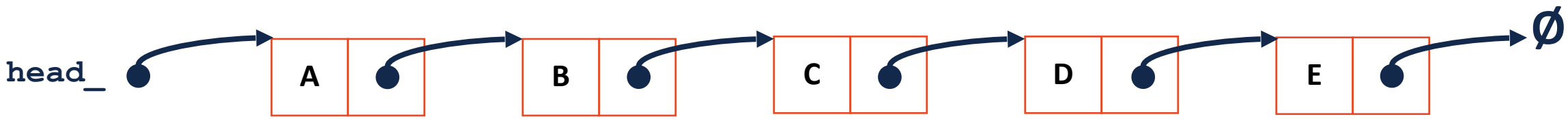


Linked List: `find(data)`

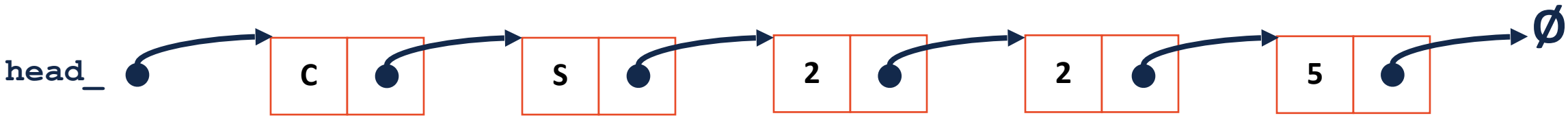


Linked List: `Remove(<parameters>)`

What input parameters make sense for remove?

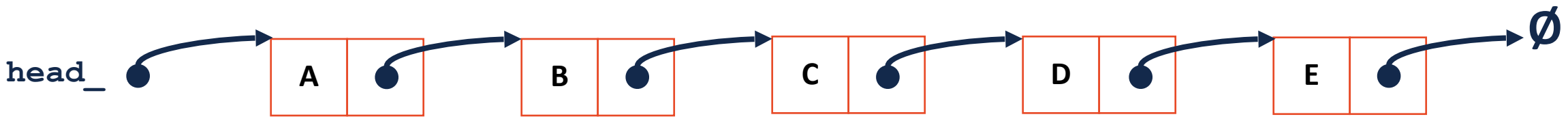


Linked List: `remove(ListNode *& node)`



List.hpp

```
103 template <typename T>
104 T List<T>::remove(ListNode *& node) {
105
106 ListNode * temp = node;
107
108 node = node->next;
109
110 T data = temp->data;
111
112 delete temp;
113
114 return data;
115
116 }
```



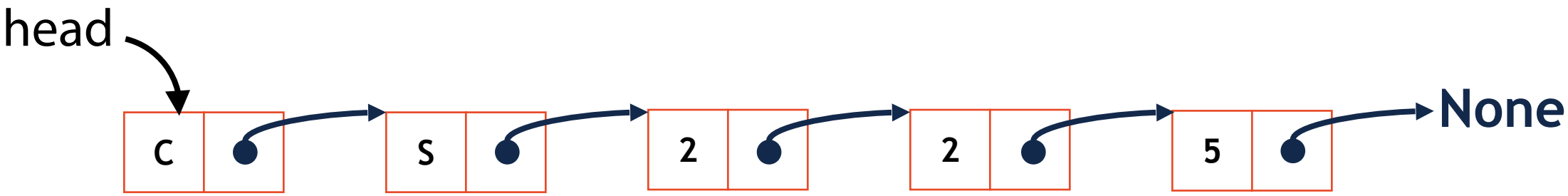
Linked List: `remove`

What is the running time to remove (if given a reference to a pointer)?

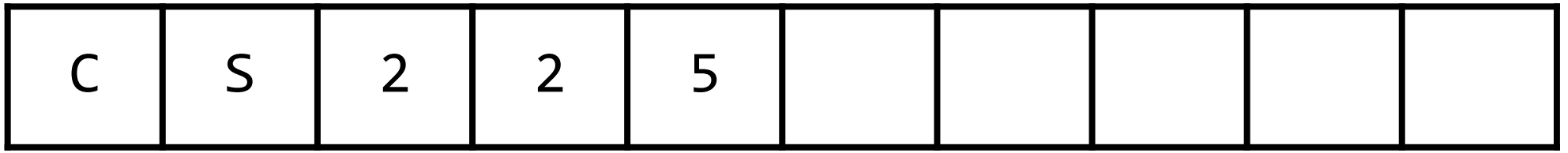
What is the running time to remove (if given a value)?

List Implementations

1. Linked List



2. Array List



List ADT

1. Insert

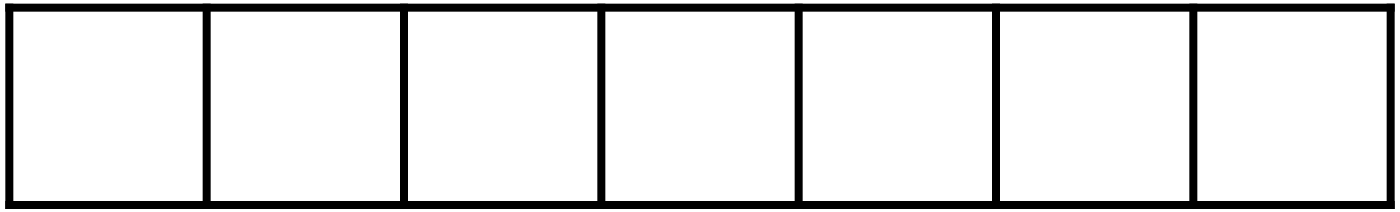
2. Delete

3. isEmpty

4. getData

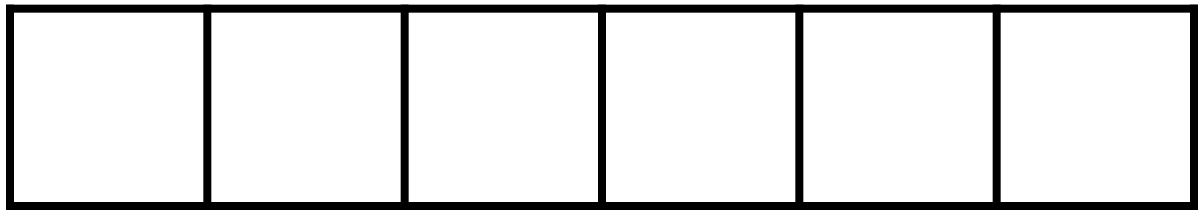
5. Create an empty list

Array List

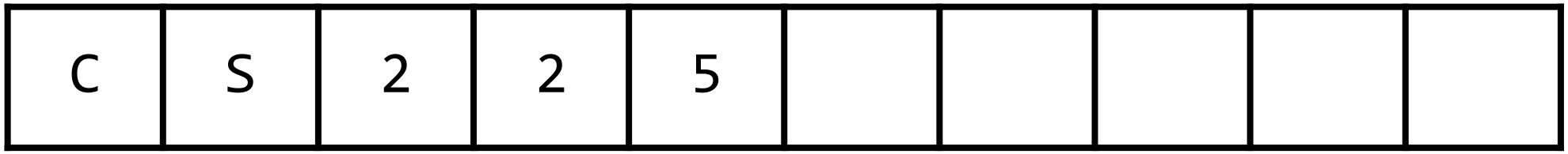


List.h

```
1  #pragma once
2
3  template <typename T>
4  class List {
5  public:
…      /* --- */
25 private:
26   T *data_;
27
28   T *size;
29
30   T *capacity;
…
       /* --- */
   };
```



Array List: [ ]



Array List: `insertAtFront(data)`