

Data Structures

Array Lists

CS 225

September 1, 2023

Brad Solomon & G Carl Evans



UNIVERSITY OF
ILLINOIS
URBANA - CHAMPAIGN

Department of Computer Science



Learning Objectives

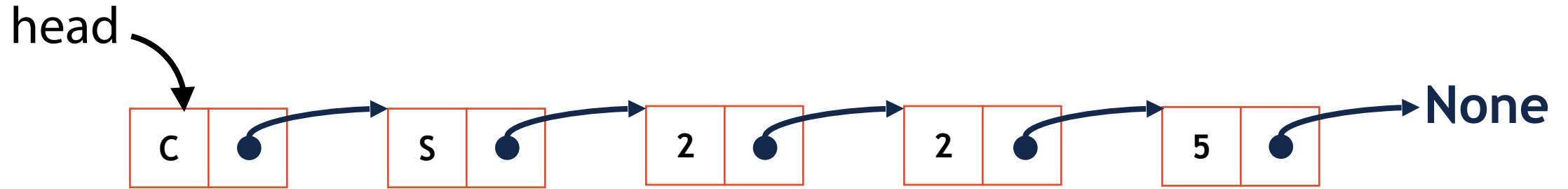
Review fundamentals of array list

Introduce array list implementations

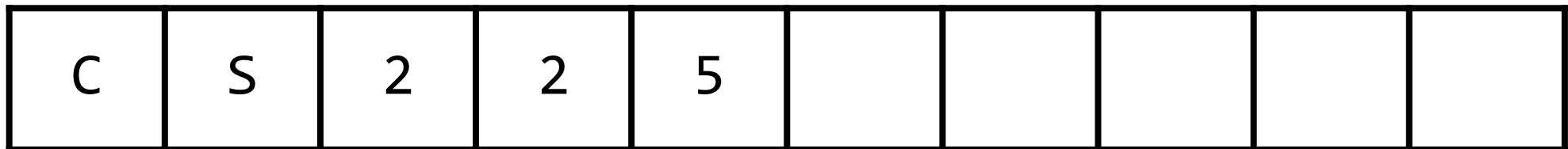
Consider extensions to lists

List Implementations

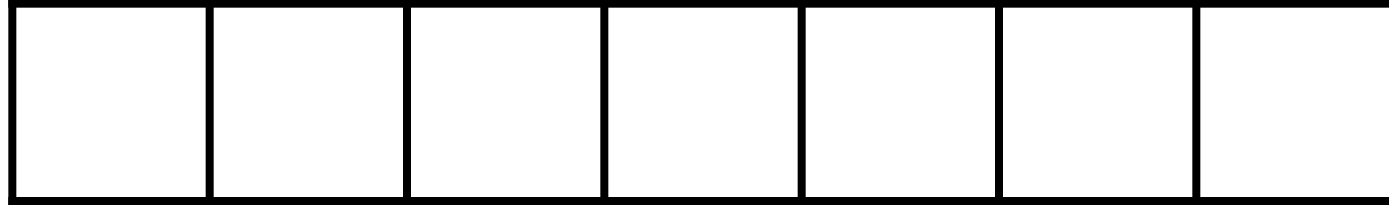
1. Linked List



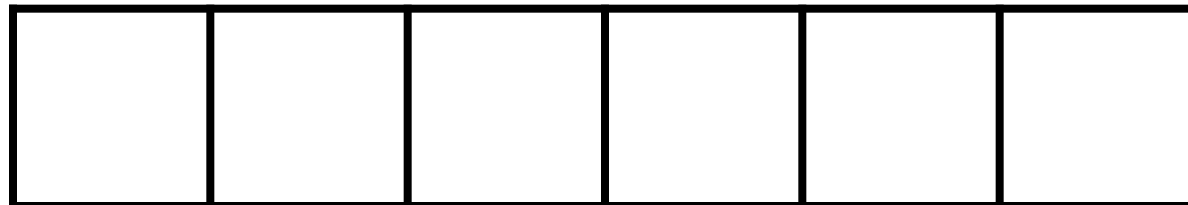
2. Array List



Array List



```
1 #pragma once
2
3 template <typename T>
4 class List {
5 public:
6     /* --- */
7
8 private:
9     T *data_;
10
11     T *size;
12
13     T *capacity;
14
15     /* --- */
16 };
```



Array List: []

c	s	2	2	5					
---	---	---	---	---	--	--	--	--	--

Array List: insertAtFront(data)

c	s	2	2	5					
---	---	---	---	---	--	--	--	--	--

Array List: `insert(data, index)`

C	S	2	2	5					
---	---	---	---	---	--	--	--	--	--

Array List: `insert(data, index)`

N	O	S	P	A	C	E
---	---	---	---	---	---	---

Resize Strategy: +2 elements every time





Resize Strategy: +2 elements every time

Resize Strategy: x2 elements every time





Resize Strategy: x2 elements every time

Array Implementation



	Singly Linked List	Array
Look up arbitrary location		
Insert after given element		
Remove after given element		
Insert at arbitrary location		
Remove at arbitrary location		
Search for an input value		

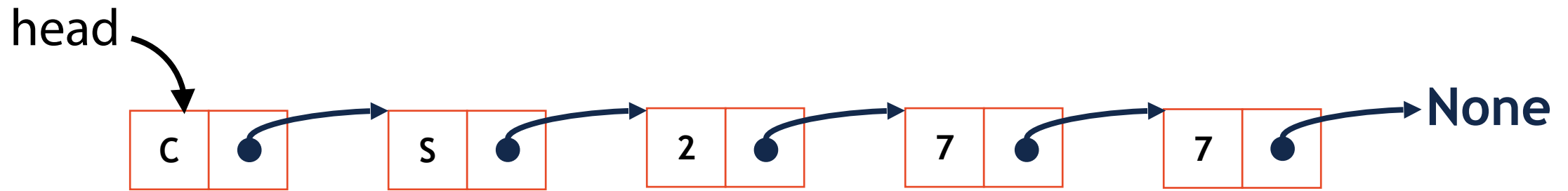
Thinking critically about lists: tradeoffs

The implementations shown are foundational.

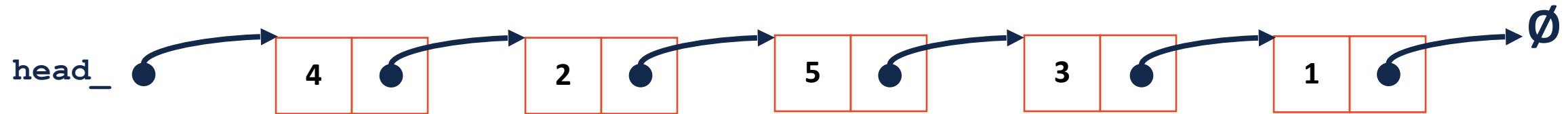
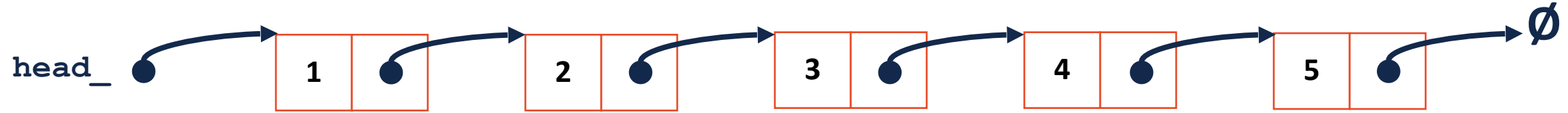
Can we make our lists better at some things? What is the cost?

Thinking critically about lists: tradeoffs

Getting the size of a linked list has a Big O of:



Thinking critically about lists: tradeoffs

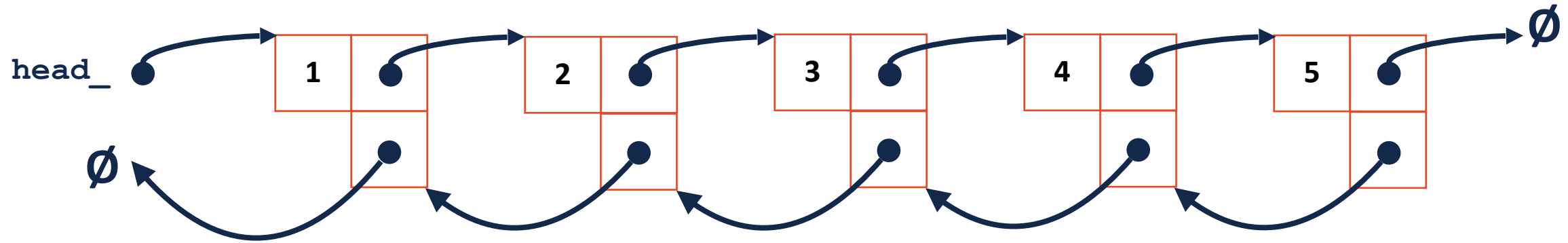


Thinking critically about lists: tradeoffs

2	7	5	9	7	14	1	0	8	3
---	---	---	---	---	----	---	---	---	---

0	1	2	3	5	7	7	8	9	14
---	---	---	---	---	---	---	---	---	----

Thinking critically about lists: tradeoffs



Thinking critically about lists: tradeoffs

When we discuss data structures, consider how they can be modified or improved!

Next time: Can we make a 'list' that is $O(1)$ to insert and remove? What is our tradeoff in doing so?