CS 225

Data Structures

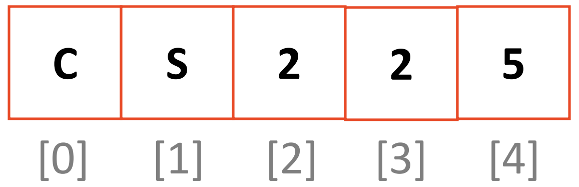
Feb. 9 – Stacks and Queues

Wade Fagen-Ulmschneider

List.h

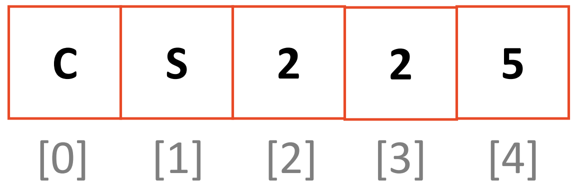
```
#ifndef LIST_H
#define LIST_H

template <typename T>
class List {
  public:
    /* ... */
  private:
    T * arr;
    unsigned capacity_; /**< Capacity of array `arr` */
    unsigned ct_;       /**< Count of data elements stored in `arr` */

};

#endif
```

Array Implementation

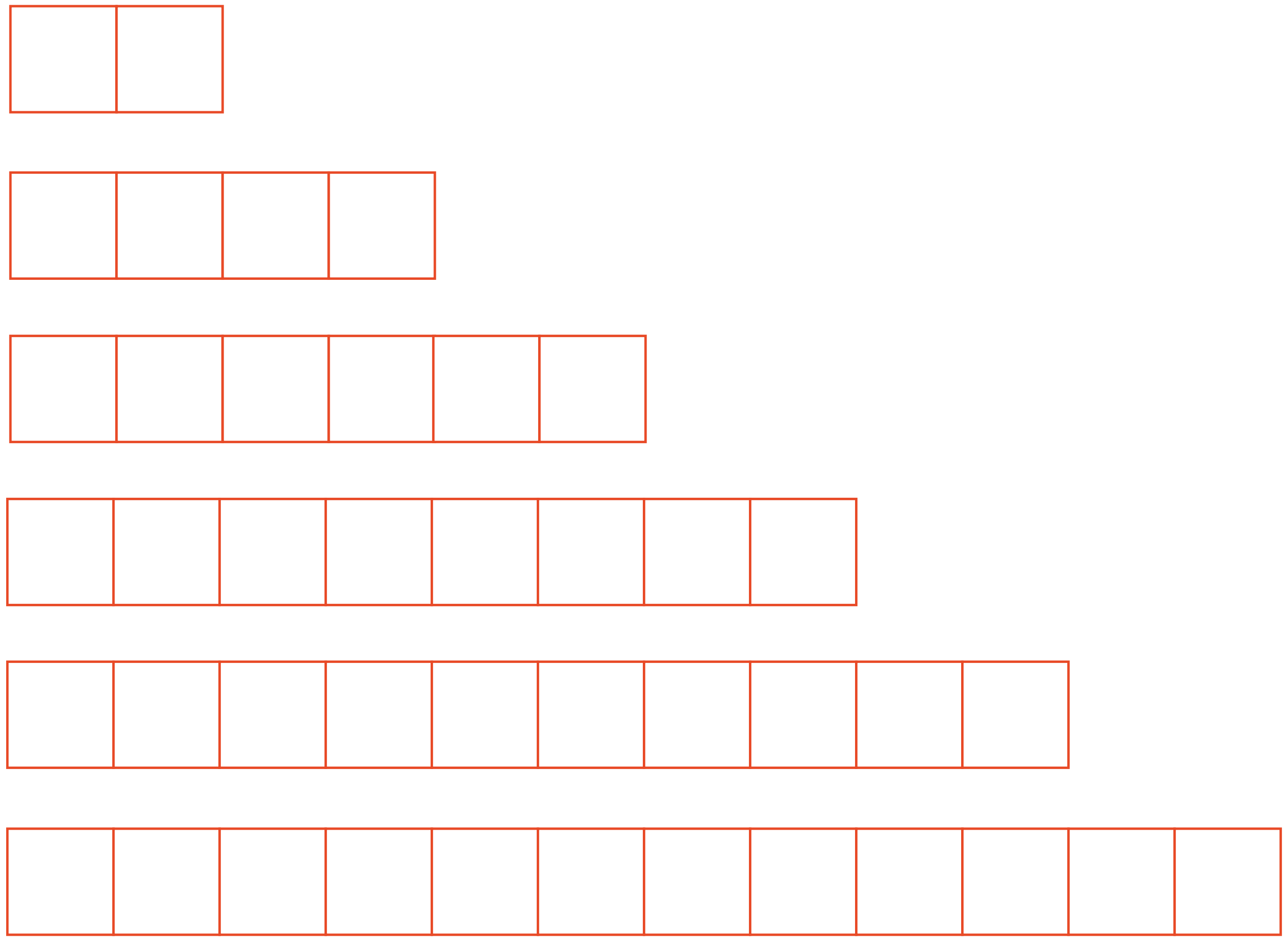


Array Implementation

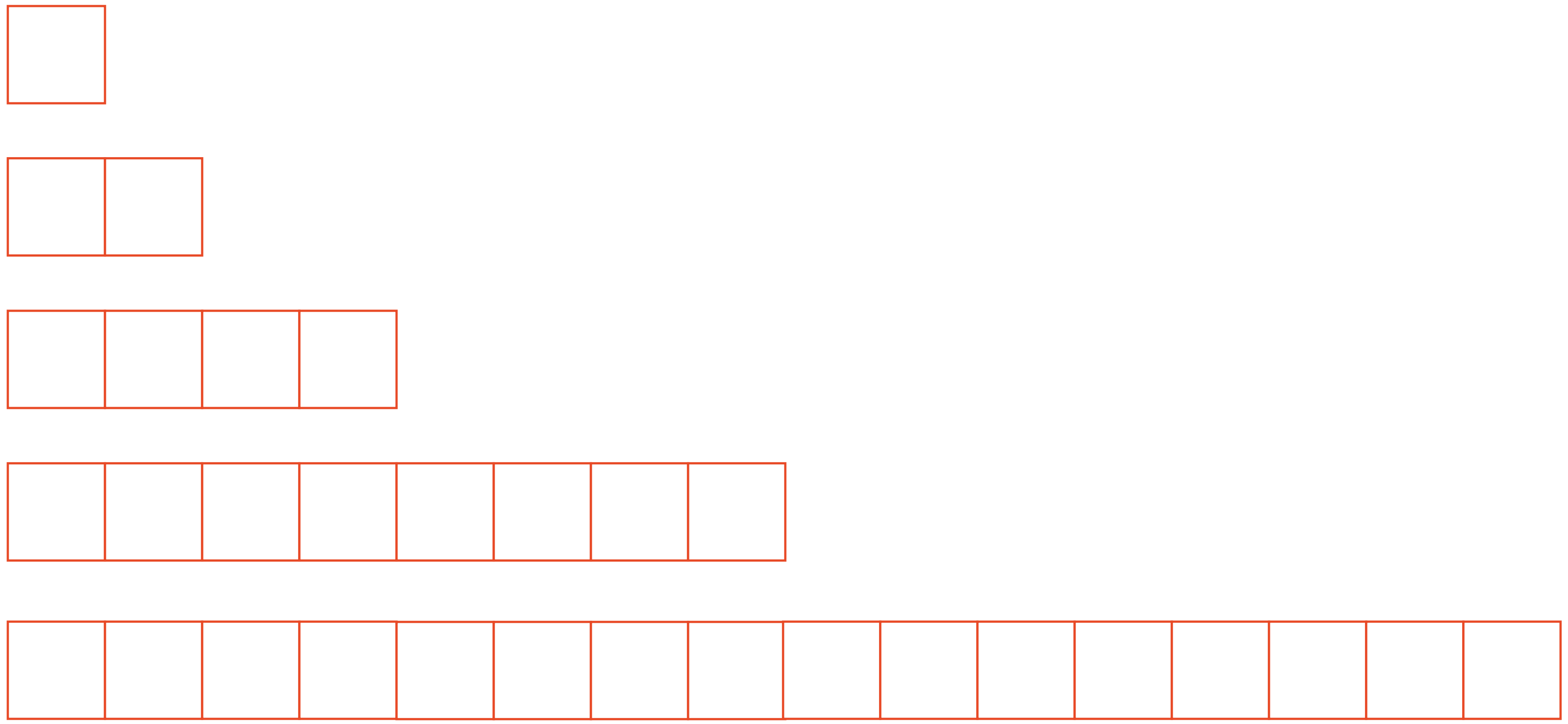
insertAtFront:



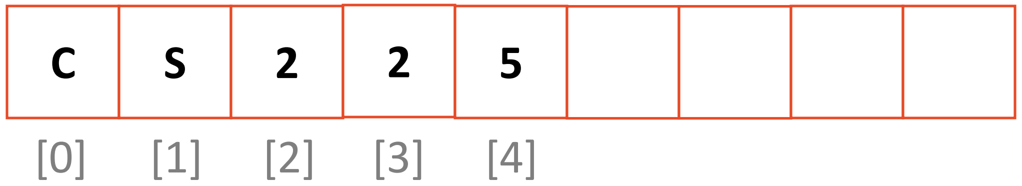
Resize Strategy – Details



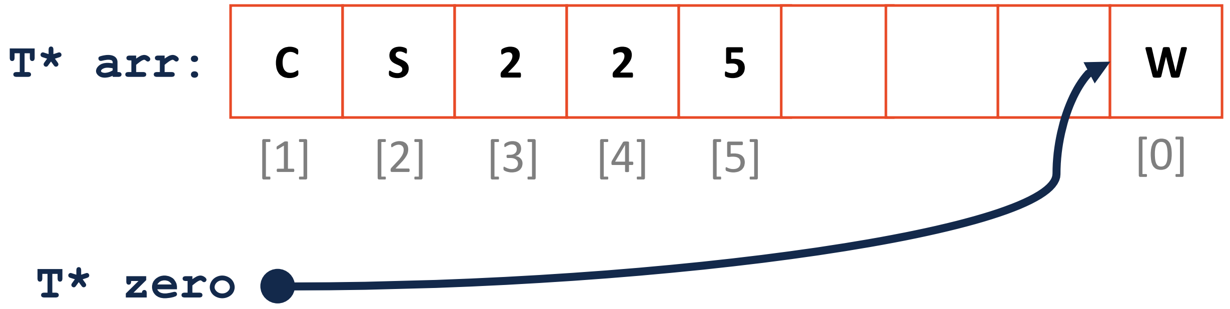
Resize Strategy – Details



Array Implementation

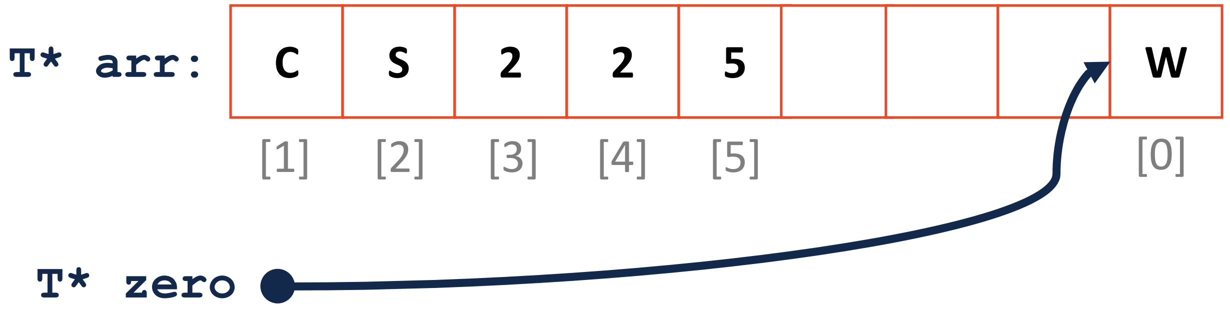


Array Implementation



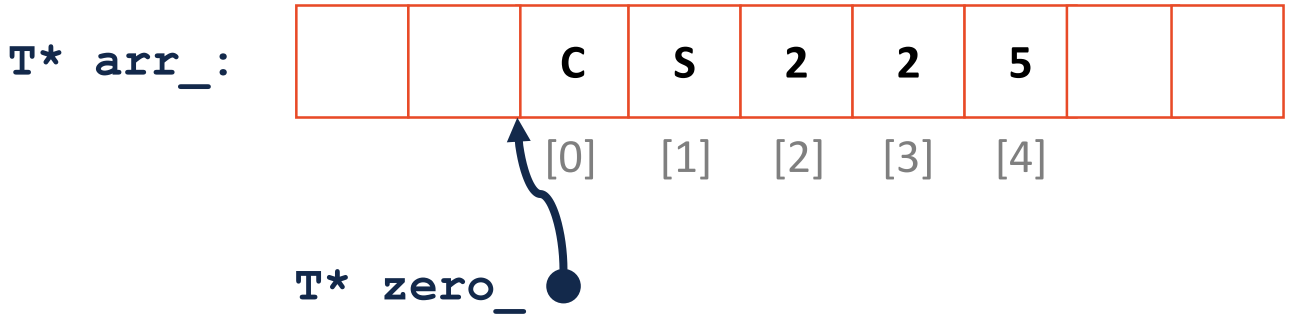
```
ListNode *& List::_get(unsigned index) const {

}
```

Array Implementation



```
ListNode *& List::_get(unsigned index) const {
  return arr_[ (zero_ - arr_) + index % capacity_ ];
}
```

Array Implementation



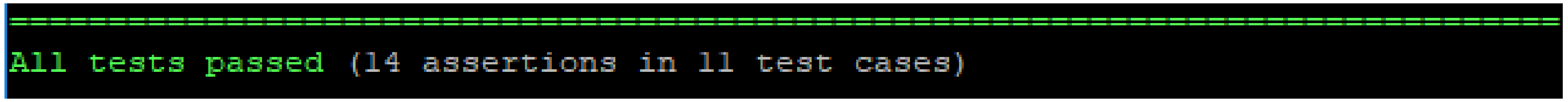
Array Implementation

	Singly Linked List	**Array**
Insert/Remove at **front**		
Insert at **given** element		
Remove at **given** element		
Insert at **arbitrary** location		
Remove at **arbitrary** location		

MP2 Updates

MP2 Updates

MP2 Updates



```
===============================================================================
All tests passed (14 assertions in 11 test cases)
```

Stack ADT

Queue ADT

Stack Implementation

```
#include "Stack.h"

template <typename T>
void Stack::push(T & t) {
  list_.add(t, 0);
}

template <typename T>
T & Stack::pop() {
  return list_.remove(0);
}

bool Stack::isEmpty() {
  return list_.isEmpty();
}
```

Implications of Design

1.

```
struct ListNode {
  T & data;
  ListNode * next;
  ListNode(T & data) : data(data), next(NULL) { }
};
```

2.

```
struct ListNode {
  T * data;
  …
```

3.

```
struct ListNode {
  T data;
  …
```

Implications of Design

	Storage by Reference	Storage by Pointer	Storage by Value
Who manages the lifecycle of the data?			
Is it possible for the data structure to store NULL?			
If the data is manipulated by user code while in our data structure, is the change reflected in our data structure?			
Is it possible to store literals?			
Speed			

Data Lifecycle

Storage by reference:

```
Sphere s;
myStack.push(s);
```

Storage by pointer:

```
Sphere s;
myStack.push(&s);
```

Storage by value:

```
Sphere s;
myStack.push(s);
```

Possible to store NULL?

Storage by reference:

```
struct ListNode {
  T & data;
  ListNode * next;
  ListNode(T & data) : data(data), next(NULL) { }
};
```

Storage by pointer:

```
T ** arr;
```

Storage by value:

```
T * arr;
```

Data Modifications

```
Sphere s(1);
myStack.push(s);

s.setRadius(42);

Sphere r = myStack.pop();
// What is r’s radius?
```

Speed