#10: List Implementations

February 15, 2021 · *G Carl Evans*

Two Basic Implementations of List:

1.

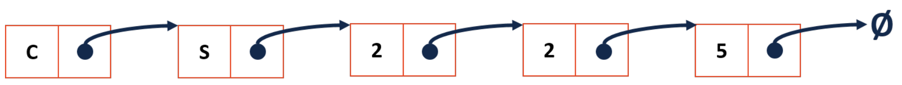
2.

A Linked List implementation of a List:

List.cpp

```
#pragma once

template <typename T>
class List {
  public:
    /* ... */

  private:
    class ListNode {
      public:
        const T data;
        ListNode * next;
        ListNode(T & data) :
           data(data), next(nullptr) { }
    };

    ListNode *head_;
    /* ... */
};
```

Implementing a basic List operation:

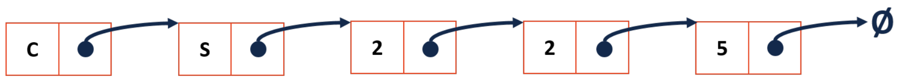


Implementing a basic List operation:

List.hpp

```
#include "List.h"

template <typename T>
void List<T>::insertAtFront(const T & d) {







}
```

Finding in a list:



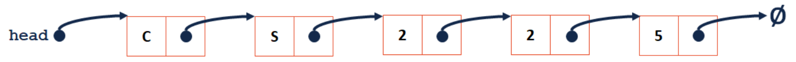
List.hpp

```
template <typename T>
typename List<T>::ListNode *&
   List<T>::_index(unsigned index) {


}




```

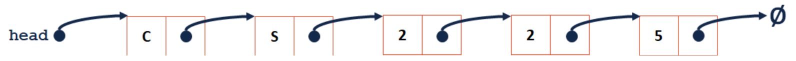
What is the return type of `_index`?

Building functionality with _index():

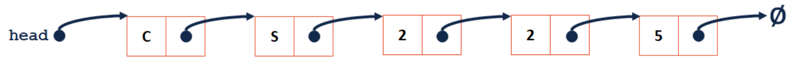
List.hpp
```
template <typename T>
T & List<T>::operator[](unsigned index) {





}
```



List.hpp
```
template <typename T>
void List<T>::insert(const T & t, unsigned index) {





}
```



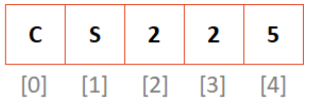
List.hpp
```
template <typename T>
T List<T>::remove(unsigned index) {





}
```



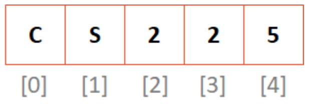
List Implementation #2: ________________

Alternate List.h
```
#pragma once

template <typename T>
class List {
  public:
      /* ... */
  private:




};
```

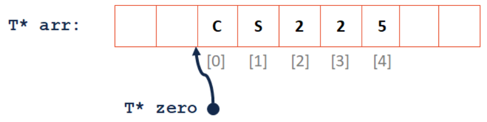
Array - Implementation Details:



1. What is the running time of `insertFront()`?



2. What is the running time of get()?



CS 225 – Things To Be Doing:

1. No lab this week.
2. mp_stickers EC **due tonight**;
3. Daily POTDs