CS 225

#11: List: Implementation (Array)

February 9, 2022 · *G Carl Evans*

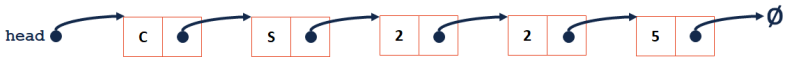
A Linked List implementation of a List:

List.cpp

```
 1  #pragma once
 2
 3  template <typename T>
 4  class List {
 5    public:
...      /* ... */
19
20    private:
21      class ListNode {
22        public:
23          const T data;
24          ListNode * next;
25          ListNode(T & data) :
                data(data), next(nullptr) { }
26      };
27
28      ListNode *head_;
...      /* ... */
...  };
```

List.hpp

```
57  template <typename T>
58  typename List<T>::ListNode *&
      List<T>::_index(unsigned index) {
59
60
61  }
62
63
64
65
```



List.hpp

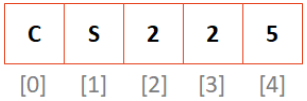
```
103  template <typename T>
104  T List<T>::remove(unsigned index) {
105
106
107
108
109  }
```

List Implementation #2: _____

Alternate List.h

```
 1  #pragma once
 2
 3  template <typename T>
 4  class List {
 5    public:
...      /* ... */
28    private:
29
30
31
32  };
```

Array - Implementation Details:



What is the running time of get()?

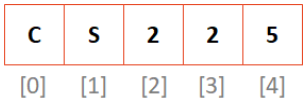
What is the running time of **`insertFront()`**?

What is the running time of **`insertBack()`**?

➔ What is our resize strategy?

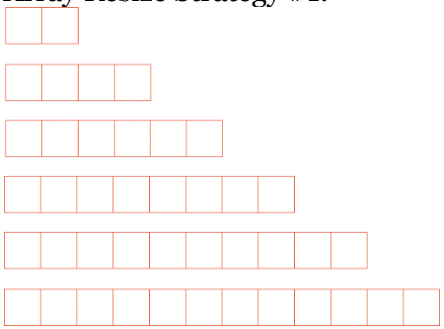
Implementation Details and Analysis:

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



➔ What is our resize strategy?

Array Resize Strategy #1:

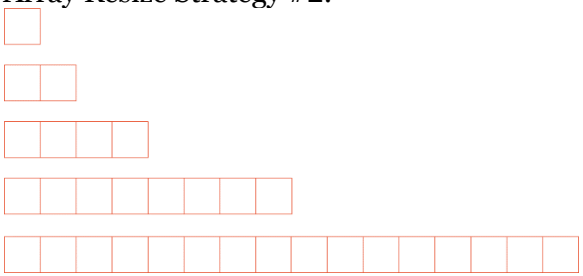


...total copies across all resizes: _____

...total number of insert operations: _____

...average (amortized) cost of copies per insert: _____

Array Resize Strategy #2:



...total copies across all resizes: _____

...total number of insert operations: _____

...average (amortized) cost of copies per insert: _____

Running Time:

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

Stack ADT

Function Name	Purpose

Queue ADT

Function Name	Purpose

CS 225 – Things To Be Doing:

1. lab_memory starts today in person
2. mp_list released later today
3. Daily POTDs