

CS 225

Data Structures

Jan 22 – Lists and List ADT

G Carl Evans



List ADT



List Implementations

1.

2.

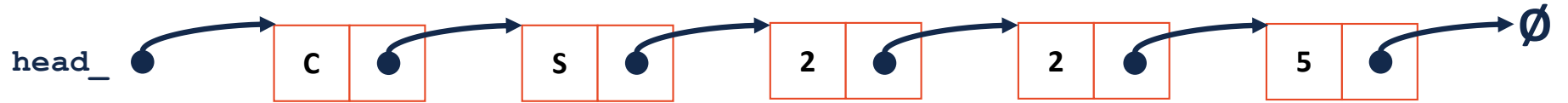
Linked Memory



List.h

```
28 struct ListNode {
29     T data;
30     ListNode * next;
31     ListNode(const T & data) : data(data), next(NULL) { }
32 };
```

Linked Memory



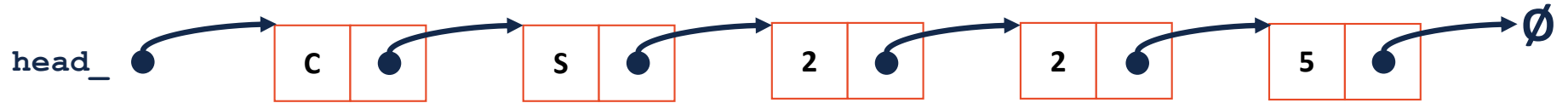
List.h

```
1 #pragma once
2
3
4 class List {
5     public:
6
7
8
9
10
11
12
13
14     private:
15
16
17
18 };
19
20
21
22
```

List.hpp

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

Linked Memory : `insertAtFront`



List.h

```
1 #pragma once
2
3 template <class T>
4 class List {
5     public:
6     /* ... */
7
8     private:
9     struct ListNode {
10         T data;
11         ListNode * next;
12         ListNode(const T & data) :
13             data(data), next(NULL) { }
14
15     };
16
17 };
18
19
20
21
22
```

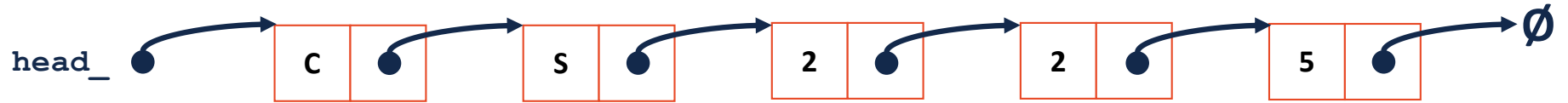
List.hpp

```
1 #include "List.h"
2
3 template <class T>
4 void List<T>::insertAtFront(const T& t) {
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 }
```



Running Time of Linked List `insertAtFront`

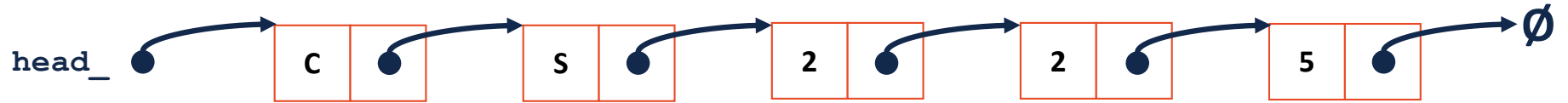
Linked Memory : `insert (data, index)`



List.hpp

```
33 ListNode *& List<T>::_index(int index) {  
34  
35  
36  
37  
38  
39  
40 }
```

Linked Memory : index

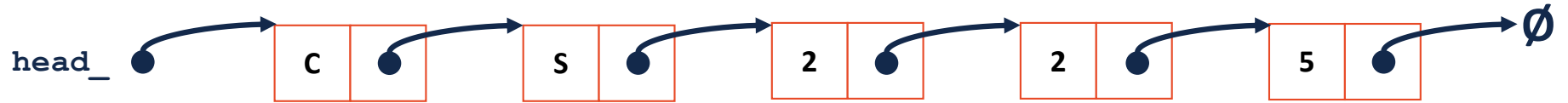


List.hpp

```
33 ListNode *& List<T>::_index(unsigned index) {  
34     return _index(index, head_)  
35 }  
}
```

```
33 ListNode *& List<T>::_index(unsigned index, ListNode *&head) {  
34  
35  
36  
37  
38  
39  
40 }
```

Linked Memory : index



List.hpp

```
// Iterative Solution:
template <typename T>
typename List<T>::ListNode *& List<T>::_index(unsigned index) {
    if (index == 0) { return head_; }
    else {
        ListNode *thru = head_;
        for (unsigned i = 0; i < index - 1; i++) {
            thru = thru->next;
        }
        return thru->next;
    }
}
```

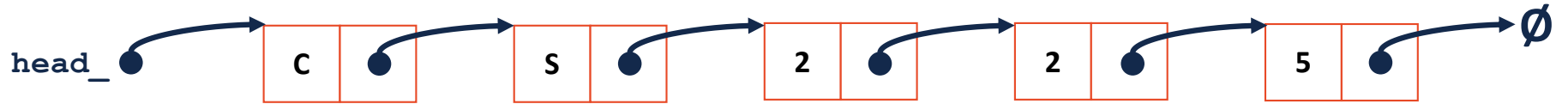



Running Time of Linked List `_index`

Recursive

Iterative

Linked Memory: **insert**



List.hpp

```
90 template <typename T>
91 void List<T>::insert(const T & t, unsigned index) {
92
93
94
95
96
97
98
99 }
```



Running Time of Linked List **insert**