Singly-Linked Lists

```
head

A  B  C  D
0  1  2  3
tail
```
A singly-linked list is a sequence of Nodes

A **Node** stores **data** and a **reference to the next node** (or null if there is no next node)
Singly-Linked Lists

Add an element to the end of the list
Singly-Linked Lists

Add an element to the end of the list

new Node(E, null)
Singly-Linked Lists

Add an element to the end of the list

new Node(E, null)
Singly-Linked Lists

Add an element to the end of the list

head

A

0

B

1

C

2

D

3

tail

new Node(E, null)

head

A

0

B

1

C

2

D

3

E

4
Remove the first element in the list
Singly-Linked Lists

Remove the first element in the list

head

A
B
C
D
E
tail

0
1
2
3
4

head

A
B
C
D
E
tail

tmp

0
1
2
3
4
Singly-Linked Lists

Remove the first element in the list
Singly-Linked Lists

Remove the first element in the list
Singly-Linked Lists

Add an element to the front of the list
Singly-Linked Lists

Add an element to the front of the list

new Node(A, head)
Singly-Linked Lists

Add an element to the front of the list

head = new Node(A, head)
How do we remove from the end of the list?
How do we remove from the end of the list?

We have to walk from the front (head) to the second last node in order to set its next field to null.

the longer the list the more costly this is! $O(n)$
Singly-Linked Lists

How do we add/remove from everywhere else in the list?

consider add(3, X)
Singly-Linked Lists

new node(X, null)
Singly-Linked Lists

```
new node(X, tmp.next)
```
Singly-Linked Lists

![Singly-Linked List Diagram]

- Head: B
- Tail: F
- Element: X
- Nodes: B, C, D, E, F, X
- Links: 0 -> 1 -> 2 -> 3 -> 4 -> 5

19
How do we add/remove from everywhere else in the list?

consider remove(3, X)