

# Arrays

create/destroy

## Can do

- put stuff in it (by index)
- get stuff out (by index)

## Can't do

- ask for length
- grow
- shrink
- check validity of indices

# List

(imagine list containing strings)

- void set(int index, string item);
- string get(int index);
- int size();
- void insertFirst(string item);
- void insertLast(string item);
- void insert(int index, string item);
- string removeFirst();

lst → insertFirst("x")

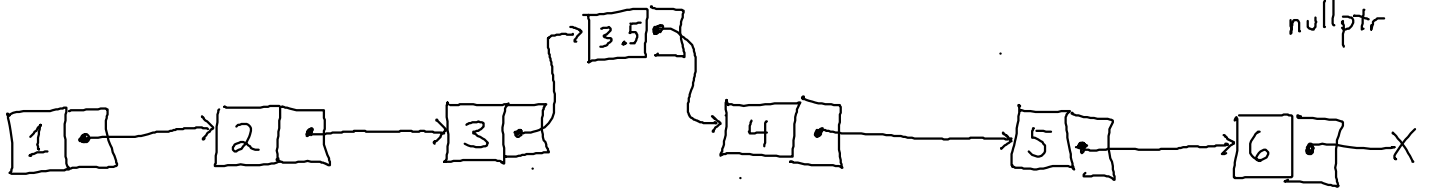
["a", "b"]



["x", "a", "b"]

ADT — abstract data type (interface to interact with a kind of data structure).

data structure — a specific implementation of algorithms that store, organize, and access data



```
template <typename T>
class LinkedListNode {
private:
    T value;
    LinkedListNode<T>* next;
};
```

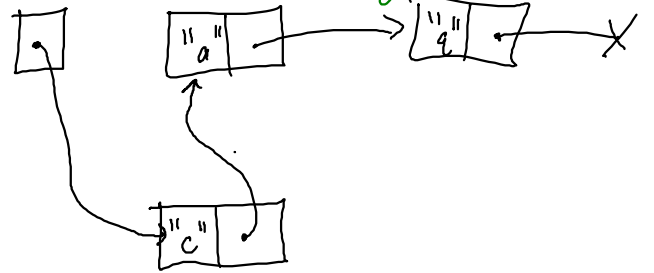
```
template <typename T> encapsulation
class LinkedList : public List<T> {
private:
    LinkedListNode<T>* head;
    int size;
};
```

invariant: "size" contains length of chain starting at head

Method insertFirst(T item):

$O(1)$  {

```
node ← new LLN
node.value ← item
node.next ← this.head
this.head ← node
this.size ← this.size + 1
```



Method size():