How does a Python List work?

Python Lists store a sequence of elements, and are quite efficient. In particular, you can add elements quite fast:

readwords1.py

```
words = []
for line in f.readlines():
  s = line.strip()
  words.append(s)
```

0.07 seconds

```
  words.insert(0, s)
```

5.8 seconds

```
  words = words + [s]
```

170 seconds

Understanding your data structures and how they implement different operations is important!

Arrays

On the hardware level, computers offer only one "data structure": the array.

An array is simply a contiguous sequence of memory locations, storing a fixed number of elements.

An Array ADT for Python:

- `Array(n)` Create an array for `n` elements.
- `len(a)` Return size of array `a`.
- `a[i]` Return or update element in slot `i` of array `a`.
- `for el in a:` Iterate over elements of array `a`.

Implementation available as `cs206array.py`.

How do Python Lists work?

How can Python implement a List (which can grow efficiently) using an array (which cannot change its size at all)?

First attempt: Increase size of array every time we append an element.

readwords2.py

Second attempt: Use an array that is larger than the required size. Only part of the array is used. We maintain the size (number of used slots) separate from the capacity (number of slots of the array).

readwords3.py

Elements need to be copied only when the size becomes equal to the capacity.

This technique is efficient if the spare capacity is proportional to the current size.

How Python lists work

Python Lists are implemented in exactly this way. We can see how their memory usage increases at certain sizes.

Module `cs206mem` provides a function to determine the capacity of a Python list.

measure1.py

Extending a list works in the same way: If the new elements fit in the spare space, they are simply copied. If not, a new array with spare capacity is created.

Inserting and removing elements works by shifting all elements to the right or to the left.

Lists created using `[13] * 1000` and lists made by splicing are created without spare slots.