Announcements

- Midterm 1 has been graded...
- Many of you did very well!
- High scores on homework and projects balance out exam scores.
- Typically, more than 75% of students receive A's & B's in 61A.
- If you are falling behind, come to class [lecture, discussion, lab, & office hours]!
- Homework 3 due Tuesday 10/1 @ 11:59pm
- Optional Hog Contest due Thursday 10/3 @ 11:59pm

Sequences

The Sequence Abstraction

- There isn’t just one sequence class or abstract data type (in Python or in general).
- The sequence abstraction is a collection of behaviors:
  - **Length**: A sequence has a finite length.
  - **Element selection**: A sequence has an element corresponding to any non-negative integer index less than its length, starting at 0 for the first element.

There is built-in syntax associated with this behavior, or we can use functions.

A tuple is a kind of built-in sequence (demo)

Box-and-Pointer Notation

- `pair = (1, 2)`
- `pairs = [(1, 2), (3, 4)]`


Box-and-Pointer Notation
The Closure Property of Data Types

- A method for combining data values satisfies the closure property if:
  - The result of combination can itself be combined using the same method.
  - Closure is the key to power in any means of combination because it permits us to create hierarchical structures.
  - Hierarchical structures are made up of parts, which themselves are made up of parts, and so on.

Tuples can contain tuples as elements

---

Recursive Lists

Constructor:

```
def rlist(first, rest):
    """Return a recursive list from its first element and the rest."""
```

Selectors:

```
def first(s):
    """Return the first element of a recursive list s."""

def rest(s):
    """Return the rest of the elements of a recursive list s."""
```

Behavior conditions:

- If a recursive list `s` is constructed from a first element `f` and a recursive list `r`, then
  - `first(s)` returns `f`, and
  - `rest(s)` returns `r`, which is a recursive list.

---

Implementing Recursive Lists with Pairs

We can implement recursive lists as pairs. We’ll use two-element tuples to encode pairs.

---

Implementing the Sequence Abstraction

```
def len_rlist(s):
    """Return the length of recursive list s.""
    length = 0
    while s != empty_rlist:
        s, length = rest(s), length + 1
    return length
```

```
def getitem_rlist(s, i):
    """Return the element at index i of recursive list s.""
    while i > 0:
        s, i = rest(s), i - 1
    return first(s)
```

Length. A sequence has a finite length.

Element selection. A sequence has an element corresponding to any non-negative integer index less than its length, starting at 0 for the first element.
Recursive implementations