Linked Lists

Python Lists

- Stored as a variable length array underneath
- Array stores references to other objects
  - all references stored in single chunk of memory
  - each reference is the same size
  - lst[i] then knows exactly how far to jump ahead in memory to get to desired item

Python Lists (cont.)

- Underlying arrays in C are technically fixed size
    - appending items -> add to end, asking for more space if necessary
    - inserting items -> add in middle, shift everything after down
    - removing items -> get rid of reference and shift everything after up

Linked List

- Data structure
- A list
  - Supports things like get, add, insert, remove, etc.
- Different underlying implementation
  - collection of nodes that are "linked" together
  - forms a sequence of elements

Linked List - Node

- Object
- Single value in list
- Stores
  - element
  - reference to next node (self-referential)

Linked List

- Sometimes a separate class from node
- `head` - reference to first `Node` in list
  - adding to start is O(1)
- `tail` (optional) - reference to last node in list
  - makes adding to / removing from end O(1)

Linked List (cont.)

- downside:
    - access is inefficient (must traverse)
    - extra memory (store pointer to next for each node)

Doubly Linked List

- Modification
  - `Node` has additional reference to previous
- Advantages
  - `remove(n1: Node)` - more efficient (no need to traverse)
  - fast removal from end