Dynamic Array Drawbacks

• use more space than just for elements
• may not actually have time for amortization
• non-Stack/Queue insert/deletion operations unduly costly
Singly Linked Lists

• (diagrams on board/web notes/textbook)
  • box-pointer convention
• Node – contains data and pointer
• Link – pointer to a node
• singly-linked: node contains one link
• traverse/link hop – move through a list
• head – start of list
• tail – end of list
Singly Linked List Operations

• (diagrams on board)
• Inserting at head
• Inserting at tail
• Inserting anywhere else
• Deleting from head
• Deleting from anywhere else
Simple List Examples

• simplelist.1.py
  • novice C style

• simplelist.2.py
  • novice C++/Java style

• simplelist.3.py
  • more OO, internal class
    • users don’t need to know about node implementation

• simplelist.4.py
  • using __ “private” naming convention
Singly Linked List Stacks & Queues

• Guaranteed O(1) operations!
• Stacks – push/pop from head
  • what happens if push/pop @ tail?
• Queues – enqueue at tail, dequeue at head
  • what happens if enqueue at head, dequeue at tail
Circular Linked Lists

- tail points to head instead of None
- analogous to circular array buffer
- Round-robin schedulers
  - cooperative multitasking
  - old OS’s, Win 3.1, Mac before OS X