Ordered Linked List Activity

Thomas Schwarz, SJ

How to implement an Ordered Linked List

- Remarks for Python and Java programmers
  - Python and Java do not have explicit pointers
  - But an object is given by its address
  - Objects persists in memory until nobody has a reference (i.e. a pointer) to them

How to implement an Ordered Linked List

- What do we need for a node:
    - A place for the key
    - A pointer to the record
    - A pointer to the next node

How to implement an Ordered Linked List

```
class Node:
    def __init__(self, key  = None, nextN = None, data = None):
        self.key = key
        self.next = nextN
        self.data = data

    def __str__(self):
        return "Node: key={}, data={}, next={}".format(self.key,
self.data, self.next)
```

How to implement an Ordered Linked List

- An ordered linked list is given by a node with sentinel value minus infinity
- To find in an OLL:
    - Follow the next pointer until you hit a node with the key that you are looking for
    - Then follow the data link

How to implement an Ordered Linked List

- Implementing look-up

```
def find(self, key):
        currentNode = self.head
        while currentNode and currentNode.key < key:
            currentNode = currentNode.next
        if currentNode and currentNode.key == key:
            return currentNode.key, currentNode.data
        else:
            return None
```

- Question: Why do I know that at the end of the while loop, currentNode.key >= key?

How to implement an Ordered Linked List

- Implementing insert

```
def insert(self, key, data):
        current = self.head
        while current.next and key > current.next.key:
            current = current.next
        if current.next and current.next.key == key:
            print('key error: insertion failed, {}'.format(key))
            return
        newNode = Node(key, current.next, data)
        current.next = newNode
        return
```

- Why do I know that current after the while loop is the node just to the left of the insertion point?

How to implement an Ordered Linked List

- Loop Invariant:
    - key > current.next.key
- Only violated when I jump out of the while loop
- Therefore
    - current.key < key < current.next.key if current.next exists
    - current.key < key otherwise

How to implement an Ordered Linked List

- Implementing delete

```
def delete(self, key):
        current = self.head
        while current.next and key > current.next.key:
            current = current.next
        if key != current.next.key:
            return
        else:
            current.next = current.next.next
```

Task: Preparation for the homework

- Implement baby bullet nodes
    - Assume that on insert, you make the node into a baby bullet node with probability 0.5
    - if random.random( ) < 0.5: make node a baby bullet