

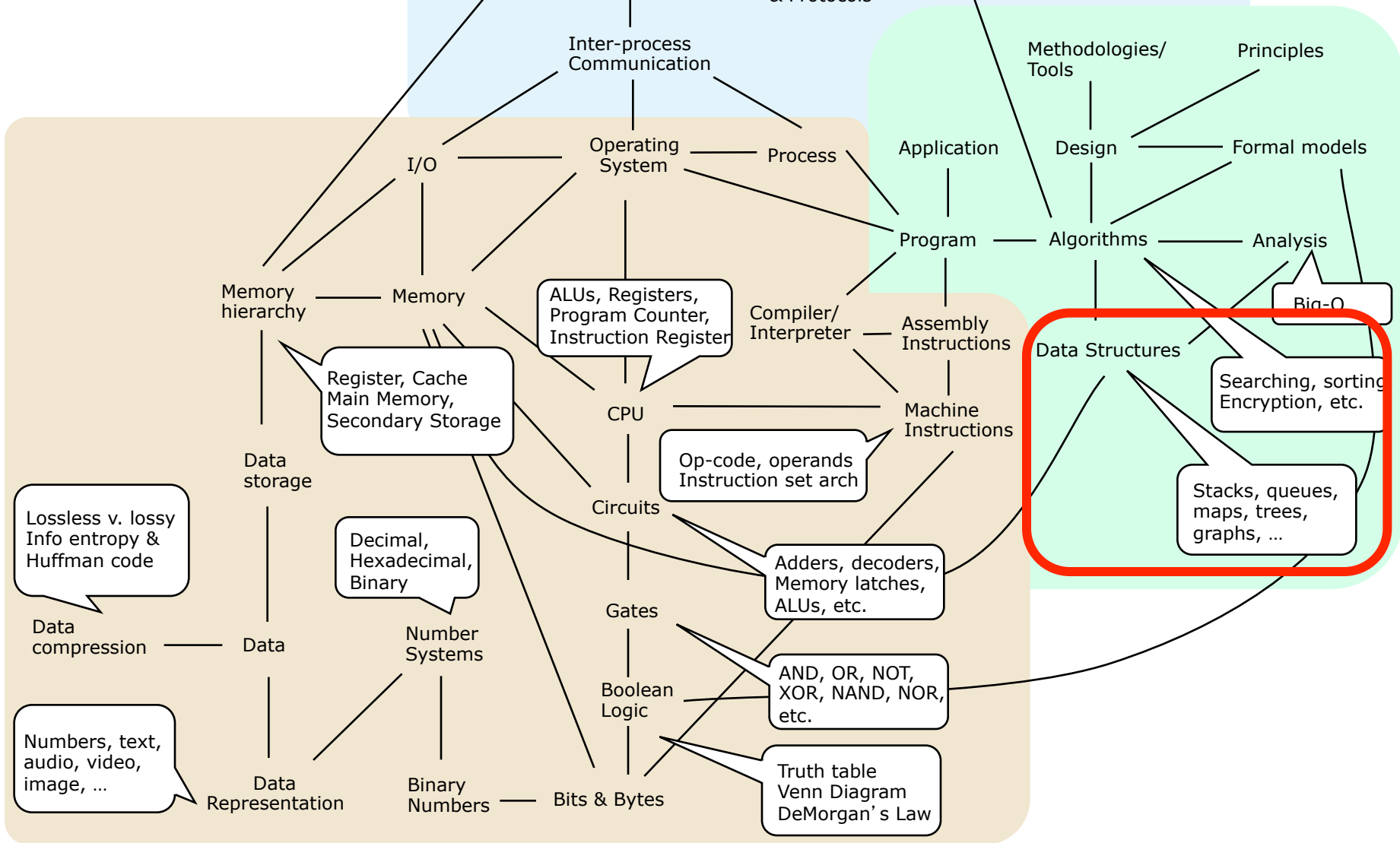
# i206: Lecture 9:

## Intro to Data Structures

Tapan Parikh  
Spring 2013

Some slides courtesy Marti Hearst, John Chuang and others

# Data Structures



# Outline

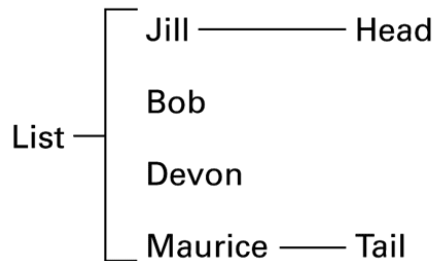
- What is a data structure
- Basic building blocks: arrays and linked lists
- Data structures (uses, methods, performance):
  - List, stack, queue
  - Dictionary
  - Tree
  - Graph

# What is a Data Structure?

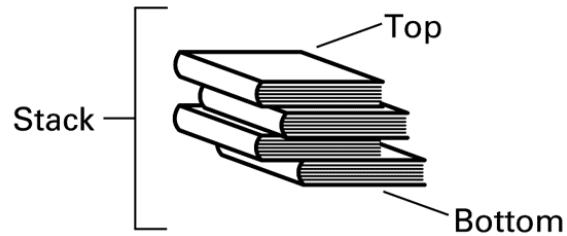
- A conceptual arrangement of data. (Brookshear)
- A systematic way of organizing and accessing data. (Goodrich & Tamassia)
- A way of storing data in a computer so that it can be used efficiently. Often a carefully chosen data structure will allow a more efficient algorithm to be used. (Wikipedia)
- Common data structures: array, list, stack, queue, dictionary, set, tree, graph, ...

# List

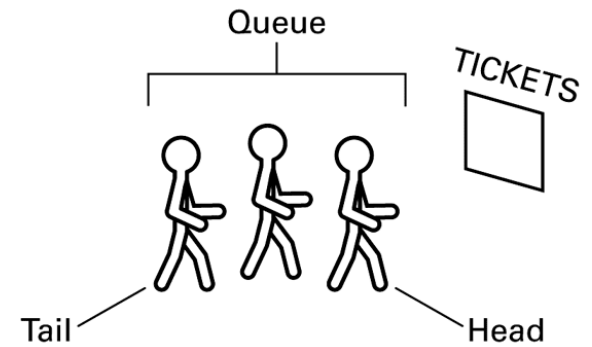
- An ordered collection of objects



a. A list of names



b. A stack of books



c. A queue of people

# Set

- An unordered collection of non-repeated objects
- Example:

```
>>> basket = ['apple', 'orange', 'apple',  
              'pear', 'orange', 'banana']  
>>> fruit = set(basket) # create a set without duplicates  
>>> fruit  
set(['orange', 'pear', 'apple', 'banana'])  
>>> 'orange' in fruit # membership testing  
True
```

# Dictionary

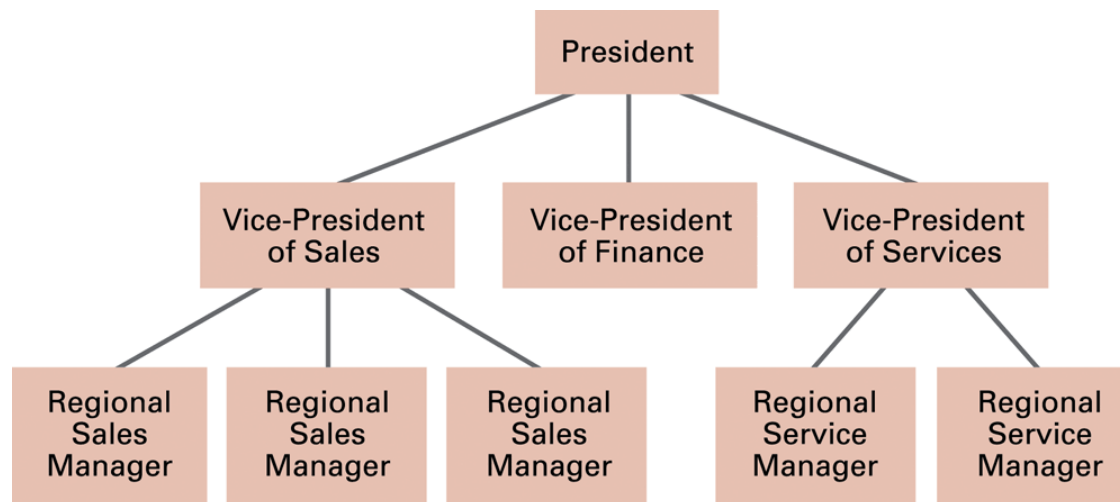
- Also known as associative array, lookup table, or map
- A searchable collection of key-value pairs
  - each key is associated with one value
- Example:

```
>>> fullname = {'chuang': 'John Chuang',  
                'i206': '206 class mailing list'}
```

```
>>> fullname['chuang']  
'John Chuang'
```

# Tree

- A collection of data whose entries have a hierarchical organization



Brookshear Figure 8.2

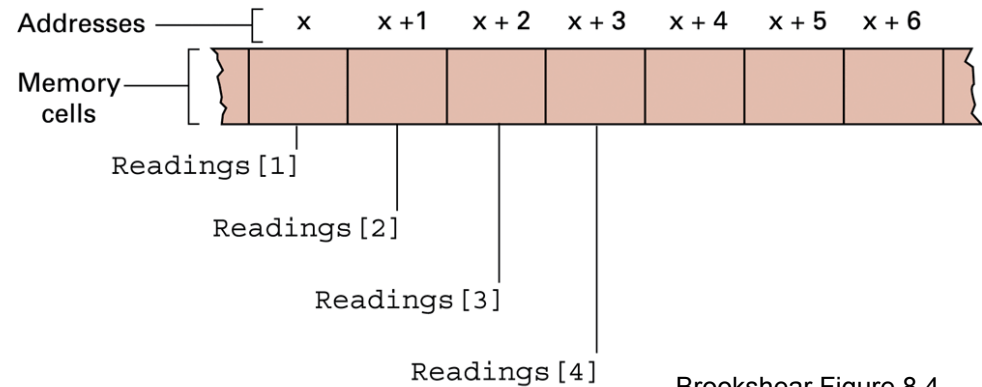


# Question

- How are these data structures implemented?
  - How are they stored in memory?

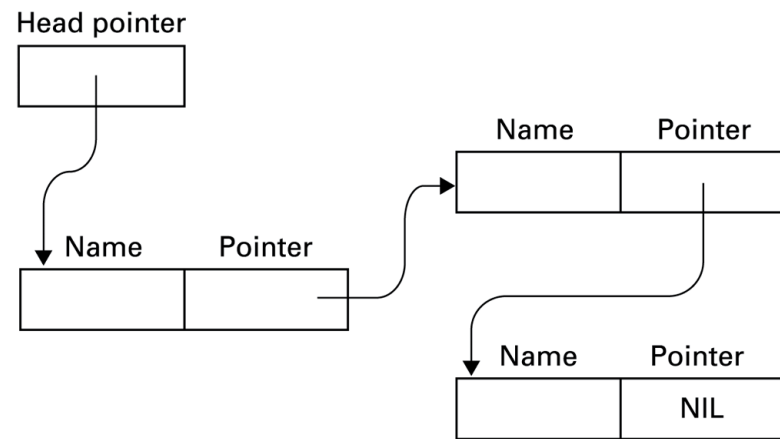
# Basic Data Structures

- Array



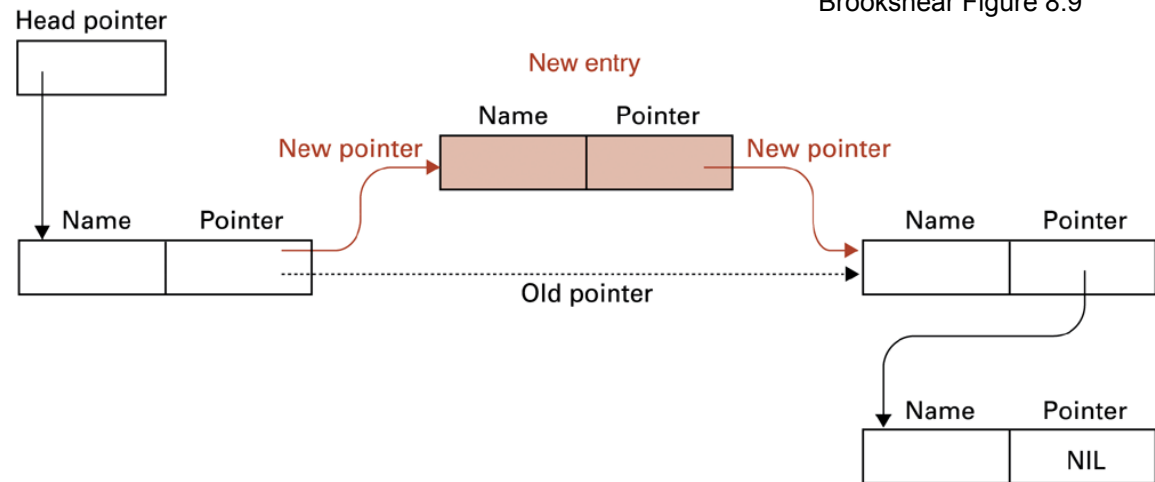
Brookshear Figure 8.4

- Linked list

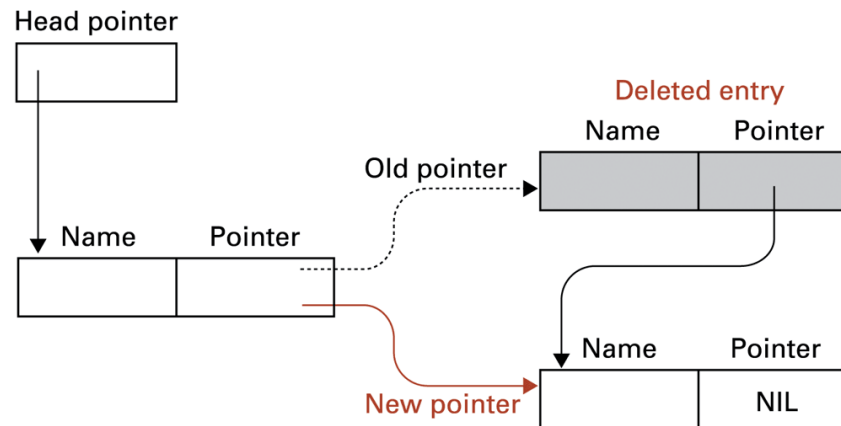


# Linked List Operations

- Insertion



- Deletion



# Array vs. Linked List

- Q: what are the tradeoffs?
  - Running time of insert, delete, lookup operations
  - Storage requirements
- Example: implement a UCB student directory

# Python Activity

- Open up a Python window, and create list of letters

- `s = "That test was easy";`
- Play around with different ways of indexing the string (which is an array of characters)

`s[2:]`

`a[:2]`

`a[0:]`

`a[-1]`

`a[-1:]`

`a[:-1]`

`a[len(a):]`

- Convert the string to how you really feel about the test