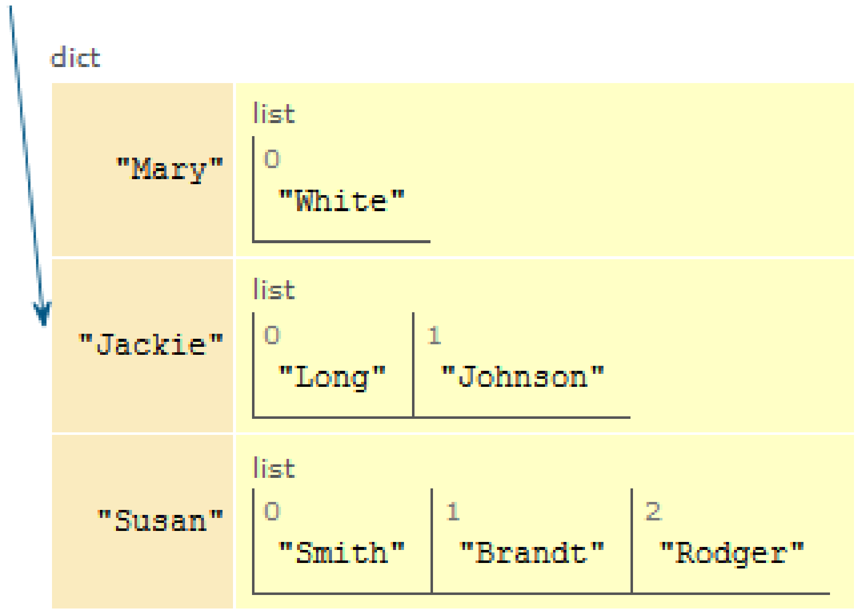
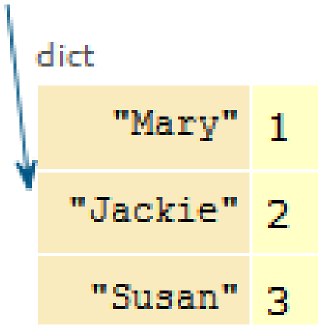
CompSci 101
Introduction to Computer Science



Oct 24, 2017

Prof. Rodger



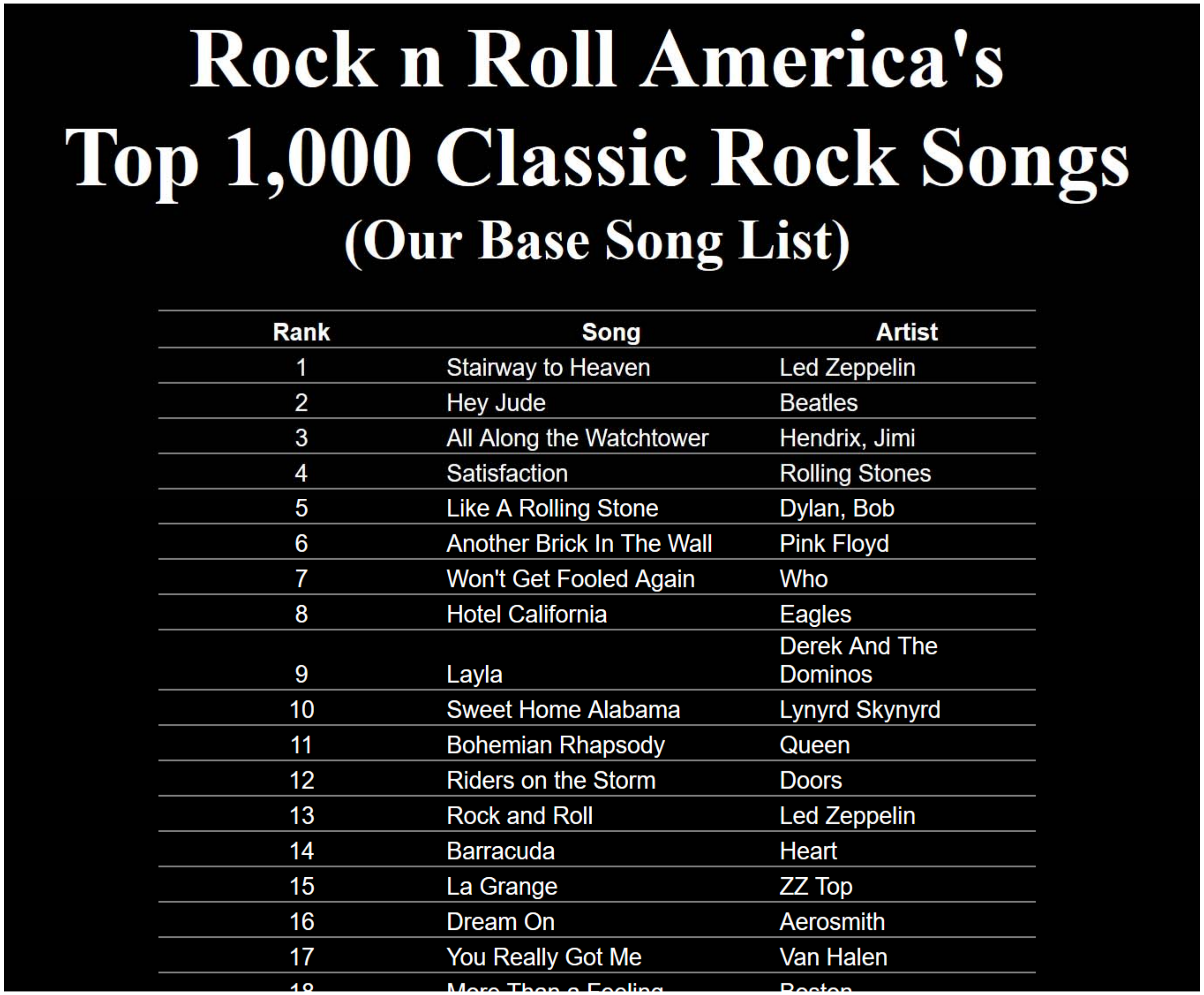
Announcements

- RQ 14 due Thursday
- Assignment 5 due Thursday
- APT 5 out, due Tues, Oct 31
- Extend Exam 1 regrade requests by Oct 26!
  - Contact gradescope if you cannot see your exam
- Lab this week! Songs and movies

- Today:
  - Finish example from last time
  - Dictionaries – a way to organize data for fast lookup

Lab this week - .csv files

- Answering questions about songs and movies

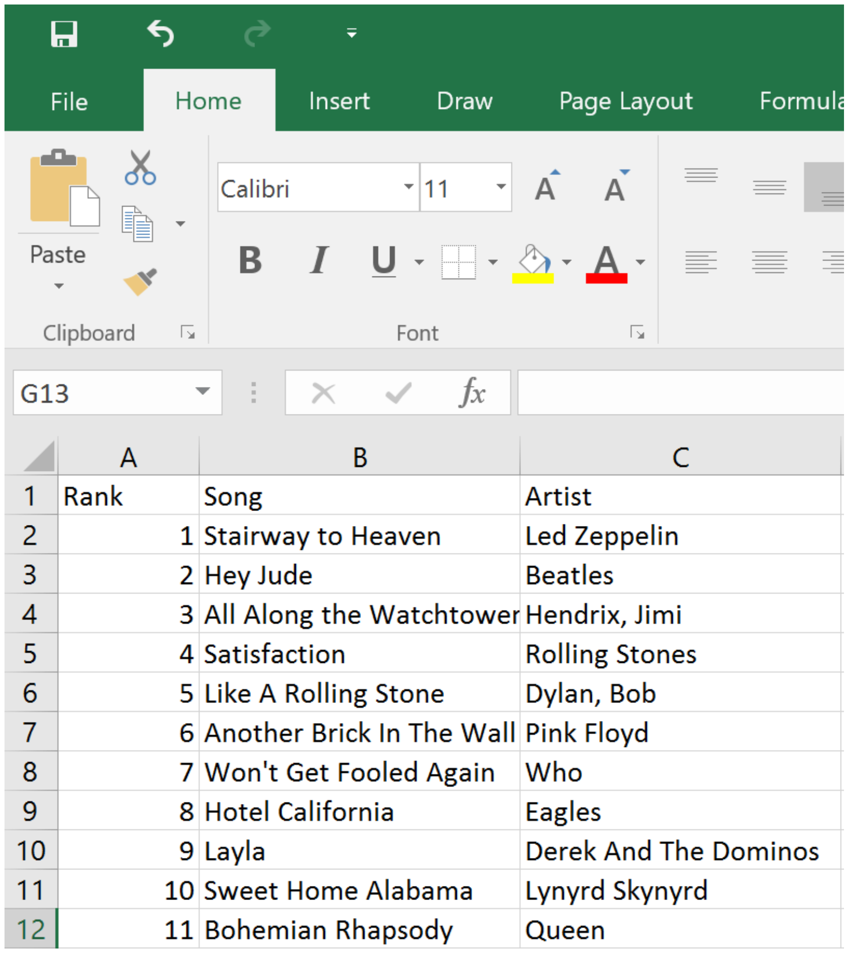


Rock n Roll America's
Top 1,000 Classic Rock Songs
(Our Base Song List)

Rank	Song	Artist
1	Stairway to Heaven	Led Zeppelin
2	Hey Jude	Beatles
3	All Along the Watchtower	Hendrix, Jimi
4	Satisfaction	Rolling Stones
5	Like A Rolling Stone	Dylan, Bob
6	Another Brick In The Wall	Pink Floyd
7	Won't Get Fooled Again	Who
8	Hotel California	Eagles
9	Layla	Derek And The Dominos
10	Sweet Home Alabama	Lynyrd Skynyrd
11	Bohemian Rhapsody	Queen
12	Riders on the Storm	Doors
13	Rock and Roll	Led Zeppelin
14	Barracuda	Heart
15	La Grange	ZZ Top
16	Dream On	Aerosmith
17	You Really Got Me	Van Halen
18	More Than a Feeling	Boston

Lab - .csv file

```
Rank,Song,Artist
1,Stairway to Heaven,Led Zeppelin
2,Hey Jude,Beatles
3,All Along the Watchtower,"Hendrix, Jimi"
4,Satisfaction,Rolling Stones
5,Like A Rolling Stone,"Dylan, Bob"
6,Another Brick In The Wall,Pink Floyd
7,Won't Get Fooled Again,Who
8,Hotel California,Eagles
9,Layla,Derek And The Dominos
10,Sweet Home Alabama,Lynyrd Skynyrd
11,Bohemian Rhapsody,Queen
12,Riders on the Storm,Doors
13,Rock and Roll,Led Zeppelin
```



	A	B	C
1	Rank	Song	Artist
2	1	Stairway to Heaven	Led Zeppelin
3	2	Hey Jude	Beatles
4	3	All Along the Watchtower	Hendrix, Jimi
5	4	Satisfaction	Rolling Stones
6	5	Like A Rolling Stone	Dylan, Bob
7	6	Another Brick In The Wall	Pink Floyd
8	7	Won't Get Fooled Again	Who
9	8	Hotel California	Eagles
10	9	Layla	Derek And The Dominos
11	10	Sweet Home Alabama	Lynyrd Skynyrd
12	11	Bohemian Rhapsody	Queen

Registration time…

- What CS courses can you take next?
  - CompSci 201
  - CompSci 216 - Everything Data
  - CompSci 230 - Discrete Mathematics

  - CompSci 230 is prereq for CompSci 330
  - CompSci 201 is prereq for many electives

LAST TIME: Problem: Popular Name

- Given a list of names, determine the most popular first name and print that name with all of its last names.
- Input: Names are always two words, names are in a file. If multiple names are on the same line they are separated by a ":"
- Output: Most popular first name, followed by a ":", followed by corresponding last names separated by a blank

Example Input File with 5 lines

```
Susan Smith:Jackie Long:Mary White
Susan Brandt
Jackie Johnson:Susan Rodger:Mary Rodger
Eric Long:Susan Crackers:Mary Velios
Jack Frost:Eric Lund
```

Corresponding Output

```
Susan: Smith Brandt Rodger Crackers
```

Example – two lists

	firstNames
0	’Susan’
1	‘Jackie’
2	‘Mary’
3	‘Eric’
4	‘Jack’

	lastNames
0	[‘Smith’,‘Brandt’,‘Rodger’,‘Crackers’]
1	[‘Long’, ‘Johnson’]
2	[‘White’,’Rodger’,’Velios’]
3	[‘Long’, ‘Lund’]
4	[‘Frost’]

Now can we solve the problem?

- Compute those two lists that are associated with each other
  - List of unique first names
  - List of corresponding last names
- Compute the max list of last names
- Now easy to print the answer.
- See popular.py

This function generates the list of lists of corresponding last names

```
def correspondingLastNames(data, firstNames):
    lastNames = [ ]
    for name in firstNames:
        lastNames.append(allLastNames(data,name))
    return lastNames
```

Finish

```
maxnum = max([len(item) for item in lastNames])
print maxnum
lastIndex = [index for (index, v) in
enumerate(lastNames) if len(v) == maxnum]
print "first name with most last names is:"
```

Expanding the Problem

- Suppose we want to read from multiple data files

  names1.txt, names2.txt, names3.txt

See processFiles in popular.py

Another way – list of lists

First word in each list is a first name
The rest are last names.

0	[‘Susan’, ‘Smith’,‘Brandt’,‘Rodger’,‘Crackers’]
1	[‘Jackie’, ‘Long’, ‘Johnson’]
2	[‘Mary’, ‘White’,’Rodger’,’Velios’]
3	[‘Eric’, ‘Long’, ‘Lund’]
4	[‘Jack’, ‘Frost’]

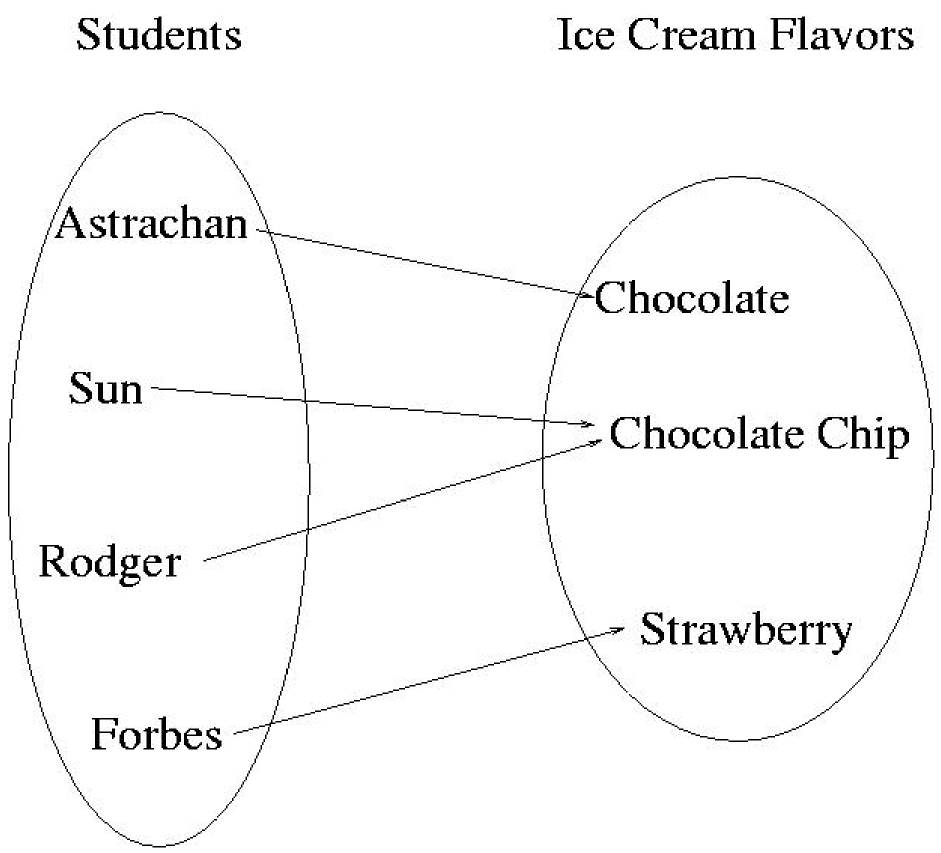
Now, a new way to organize data….

Dictionaries/Maps

- Dictionaries/maps are another way of organizing data
- Keys and Values
  - Each key maps to a value
  - Some keys can map to the same value
  - Can change the value a key maps to

Example

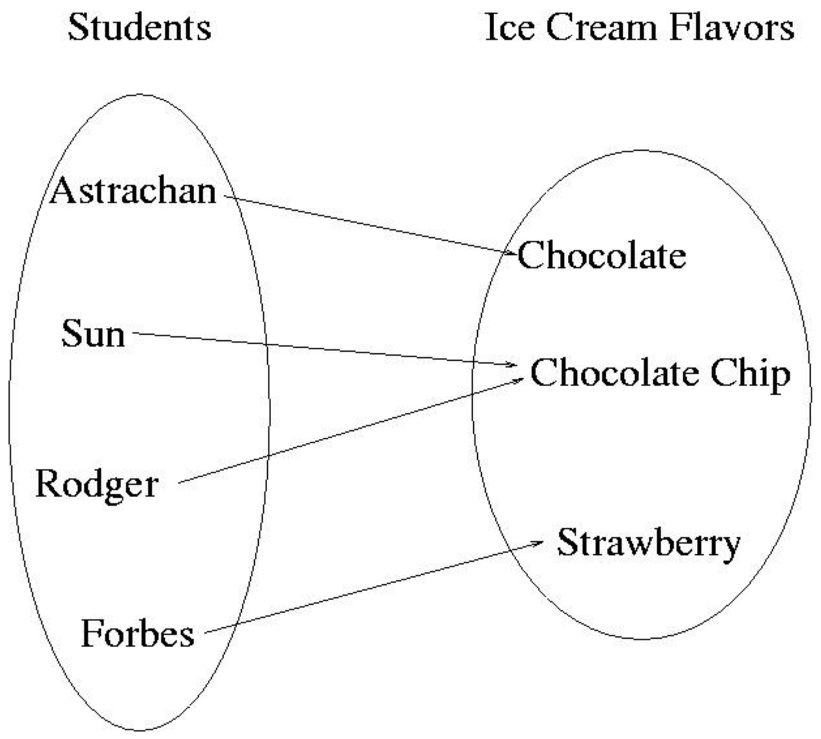
- Each student could be mapped to their favorite ice cream flavor



How is dictionary different than a list?

- List – have to search for name first
- Dictionary – each key maps to a value
- getting name (or key) is automatic! Fast!

Keys



Values

Implementing a Dictionary/Map Keys map to values

- Create Empty dictionary

  somemap = {}

- Put in a key and its value

  somemap[“Forbes”] = “Strawberry”

- Get a value for a dictionary

  value = somemap[“Forbes”]

  OR value = somemap.get(“Forbes”, “default”)

- Change a value for a dictionary

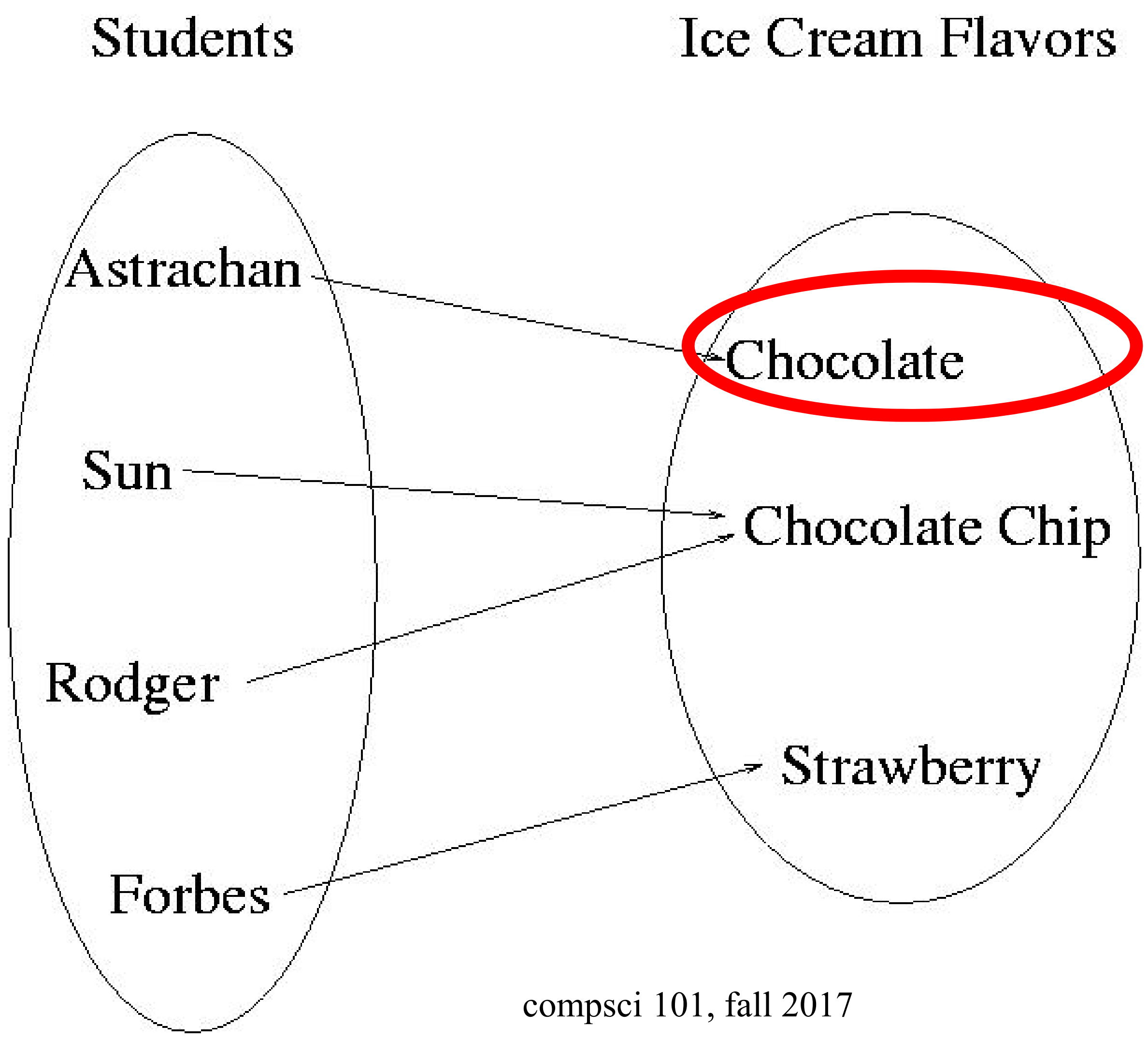
  somemap[“Forbes’] = “Chocolate”

More on using a Dictionary/Map

- Get all the keys (as a list)
  - `listKeys = somemap.keys()`
- Get all the values (as a list)
  - `listValues = somemap.values()`
- Other methods
  - `clear` – empty dictionary
  - `items` – return (key,value) pairs
  - `iteritems` – return (key,value) pairs more efficiently, *iterator – must use with for*
  - `update` – update with another dictionary

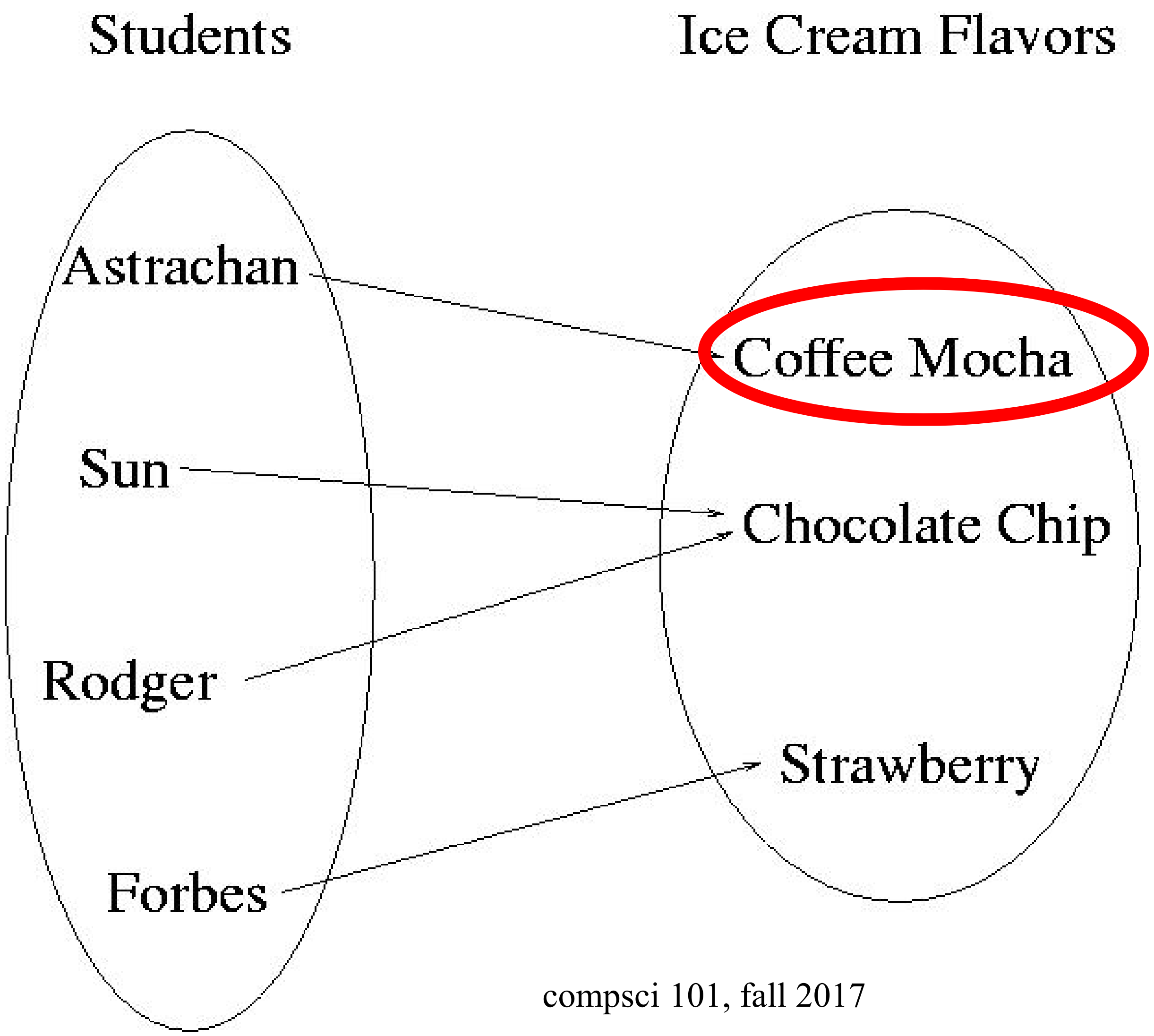
Change Astrachan’s value

somemap[“Astrachan”] = Coffee Mocha

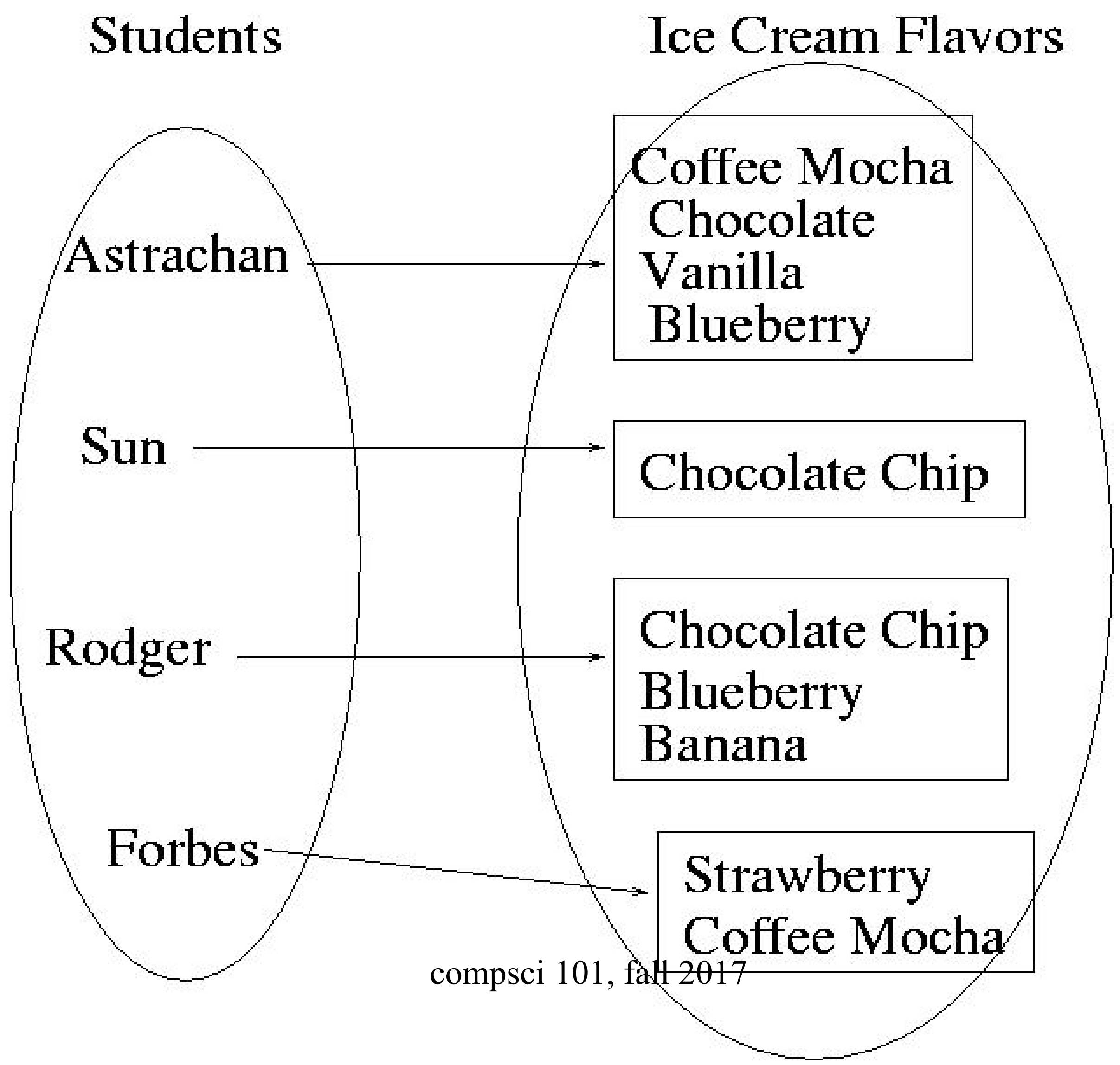


Change Astrachan’s value

somemap[“Astrachan”] = Coffee Mocha



Value could be a set or list



Simple dictionary
bit.ly/101f17-1024-1

More simple dictionaries
bit.ly/101f17-1024-2

Back to Popular Name Problem:

- Given a list of names, determine the most popular first name and print that name with all of its last names.
- Input: Names are always two words, names are in a file. If multiple names are on the same line they are separated by a ":"
- Output: Most popular first name, followed by a ":", followed by corresponding last names separated by a blank

Example Input File with 5 lines

```
Susan Smith:Jackie Long:Mary White
Susan Brandt
Jackie Johnson:Susan Rodger:Mary Rodger
Eric Long:Susan Crackers:Mary Velios
Jack Frost:Eric Lund
```

Corresponding Output

```
Susan: Smith Brandt Rodger Crackers
```

Use a dictionary/map
www.bit.ly/101f17-1024-3

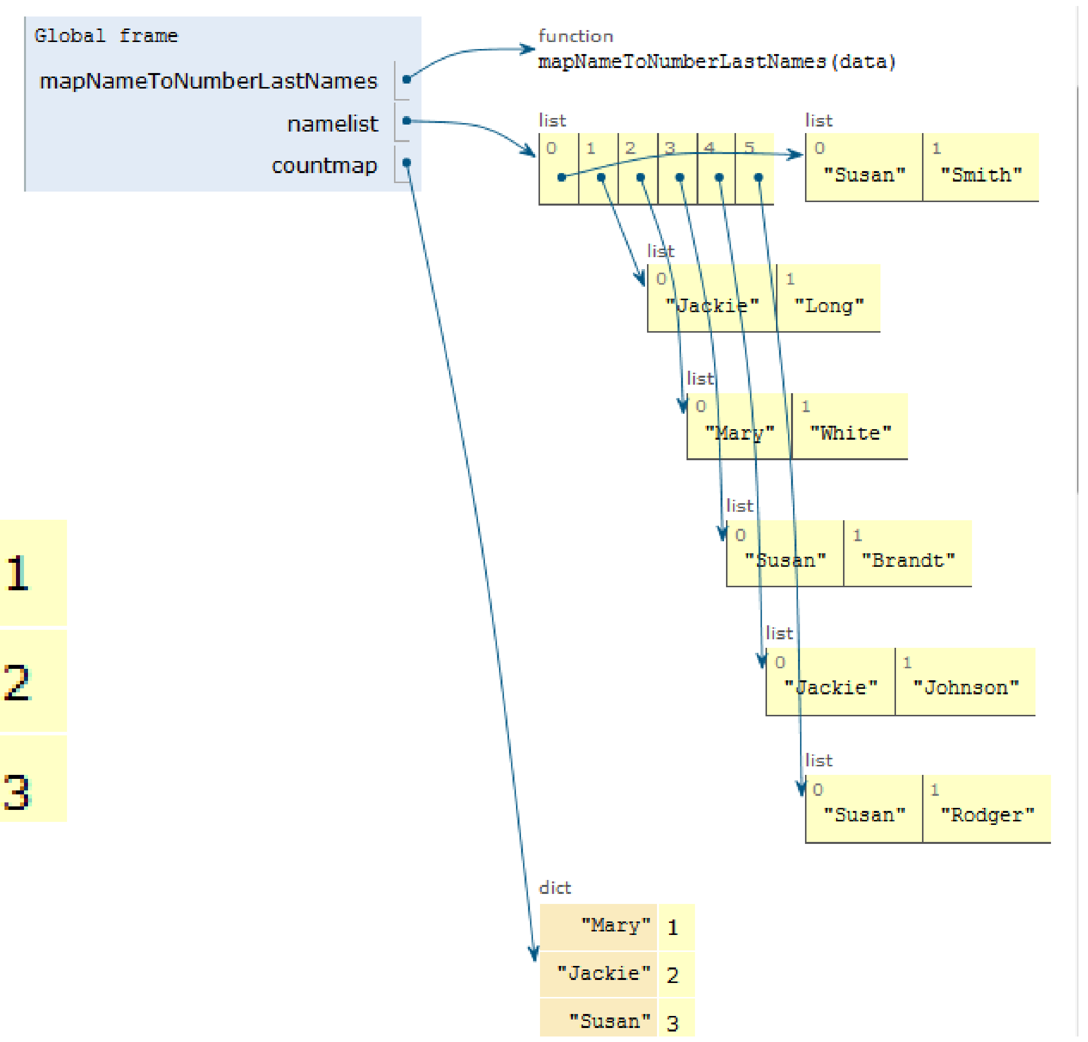
- Map first names to count of corresponding last names

def **mapNameToNumberLastNames(data):**

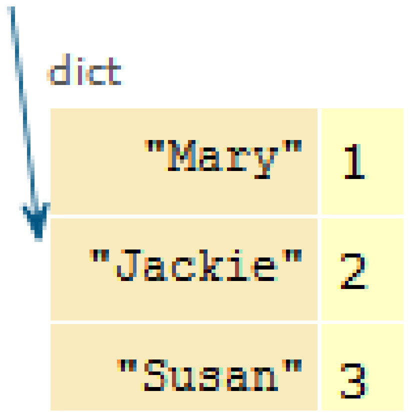
Use a dictionary/map

- popularMap.py

Trace example with Python Tutor see popularMapSolnSmall.py



"Mary"	1
"Jackie"	2
"Susan"	3



"Mary"	1
"Jackie"	2
"Susan"	3

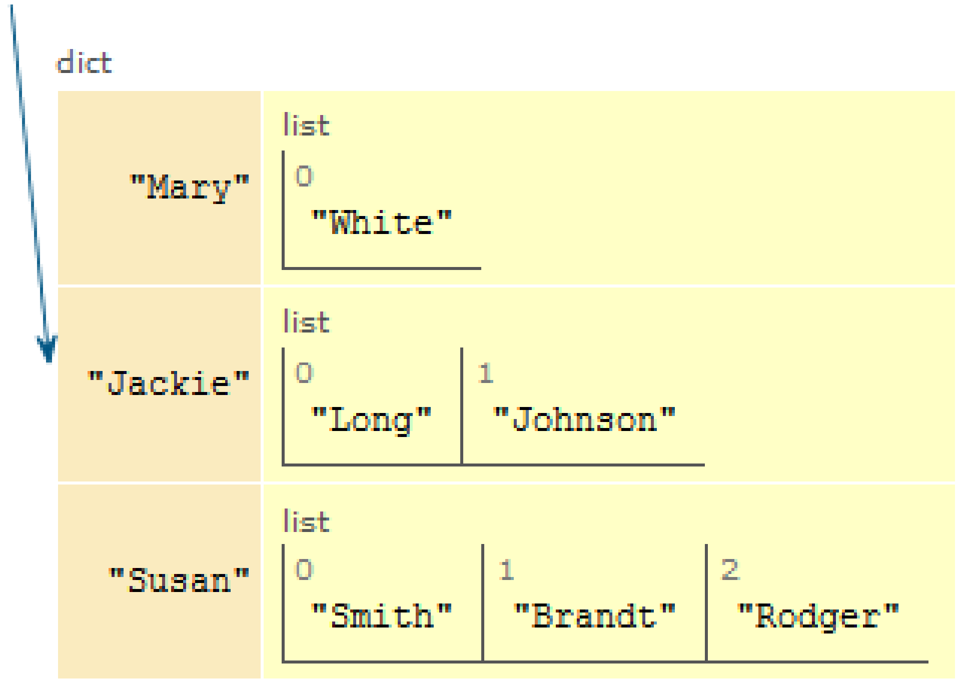
Use a dictionary/map
www.bit.ly/101f17-1024-4

- Map first name to list of corresponding last names

def **mapNameToLastNames(data):**

Trace through example with Python Tutor

- See the small example popularMapSolnSmall.py



Use dictionary of first names mapped to corresponding last names

- How do you find the most popular first name?

Use a dictionary/map

www.bit.ly/101f17-1024-5

- Map first name to set of corresponding last names

def **mapNameToSetLastNames(data):**

Compare

- Using two parallel lists?
- Using one dictionary/map
- Which dictionary is most useful to solve the most popular name problem?