

CompSci 101

Introduction to Computer Science

dict

"Mary"	list	0	"White"
"Jackie"	list	0	"Long" 1 "Johnson"
"Susan"	list	0	"Smith" 1 "Brandt" 2 "Rodger"

Nov. 3, 2016

Prof. Rodger

dict

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

Announcements

- No Reading or RQ until after exam
- Assignment 6 out today
- APT 7 due Tues
- APT Quiz 2 – Sunday - Tuesday
 - Pick 3 hours to take it
- Today:
 - Finish from last time
 - Dictionaries – a way to organize data for fast lookup

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
```

Answer questions about bit.ly/101f16-1103-1

- Printing first names with corresponding last names
- Reading data from files

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’]

ACM Turing Award Winners 2015



The screenshot shows the homepage of the ACM Turing Award Winners 2015 website. At the top left is the ACM logo and a link to 'MORE ACM AWARDS'. In the center is a large image of a man's face with the text 'A.M. TURING AWARD' overlaid. At the top right are a search bar and a 'TYPE HERE' input field. Below the search bar is a grid of 24 small portrait photos of Turing award winners. A navigation bar at the bottom includes links for 'ALPHABETICAL LISTING', 'YEAR OF THE AWARD', and 'RESEARCH SUBJECT'.



2015 AWARD WINNERS:

**Whitfield Diffie and
Martin Hellman**

Cryptography Pioneers Receive 2015 ACM A.M. Turing Award

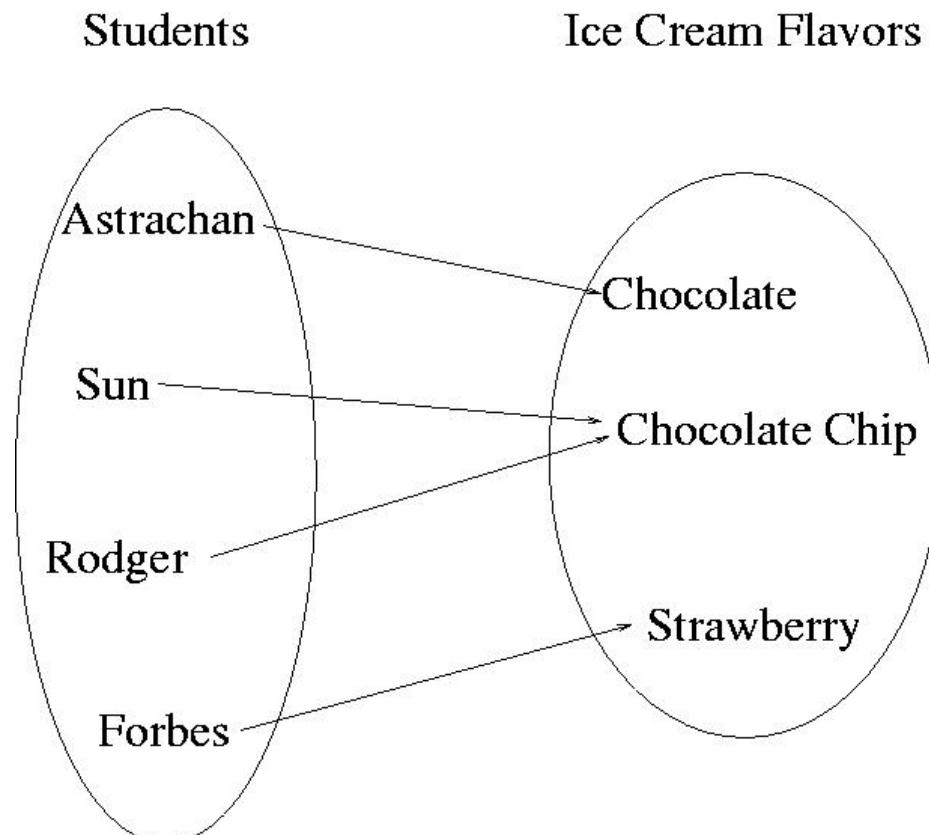
Whitfield Diffie, former Chief Security Officer of Sun Microsystems and Martin E. Hellman, Professor Emeritus of Electrical Engineering at Stanford University, are the recipients of the 2015 ACM A.M. Turing Award, for critical contributions to modern cryptography. The ability for two parties to communicate privately over a secure channel is fundamental for billions of people around

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

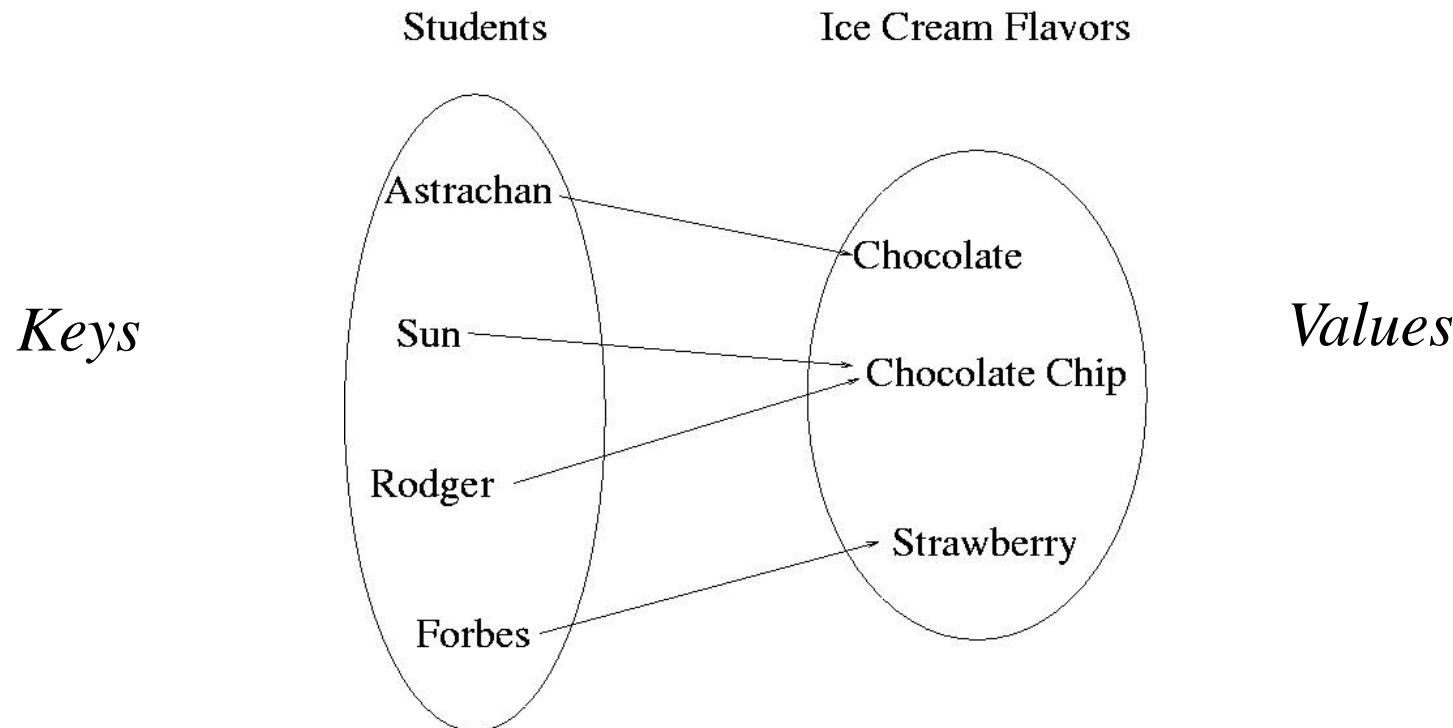
Example

- 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!



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

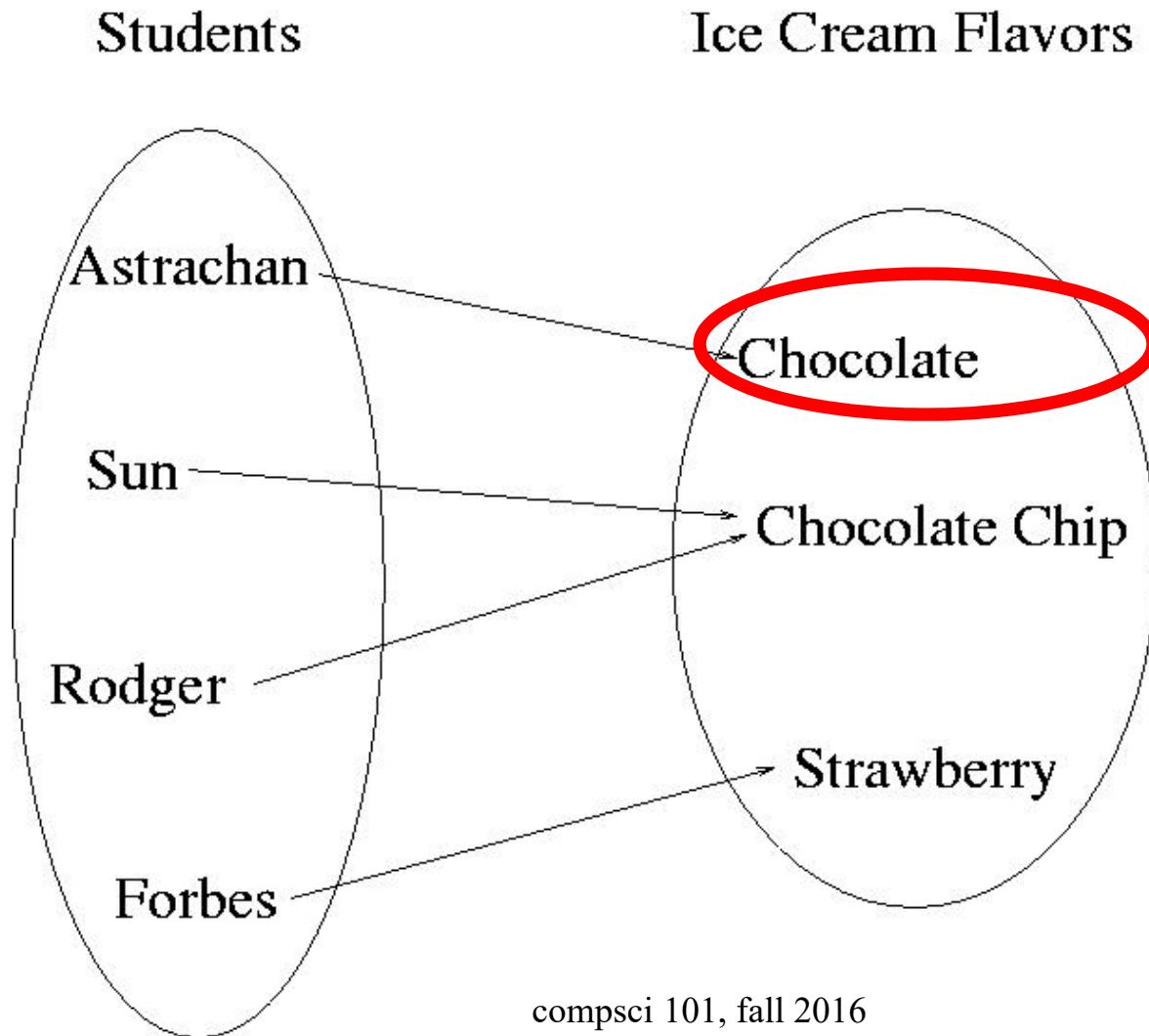
 somemap[“Forbes’] = “Chocolate”

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

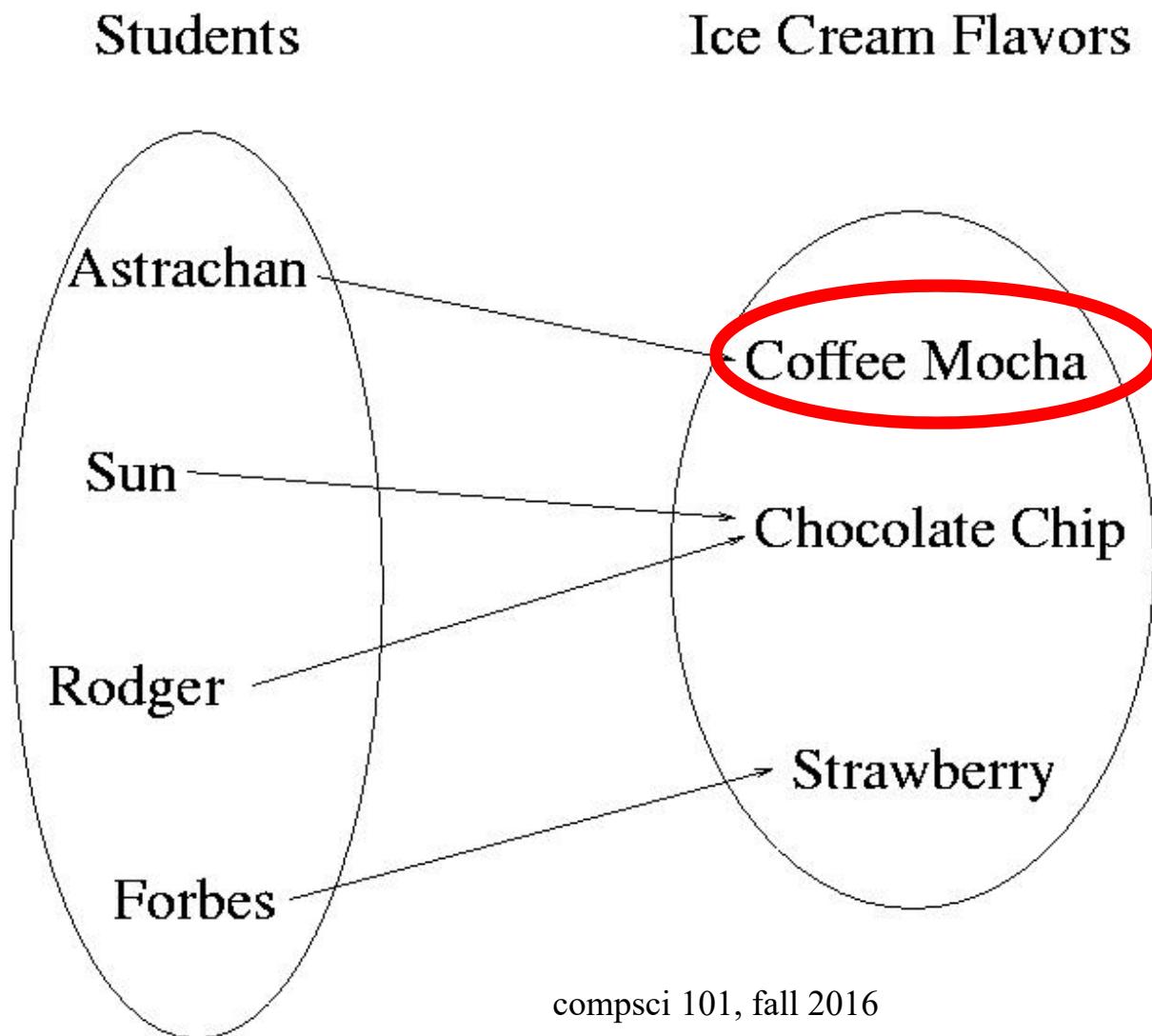
Change Astrachan's value

`somemap["Astrachan"] = Coffee Mocha`

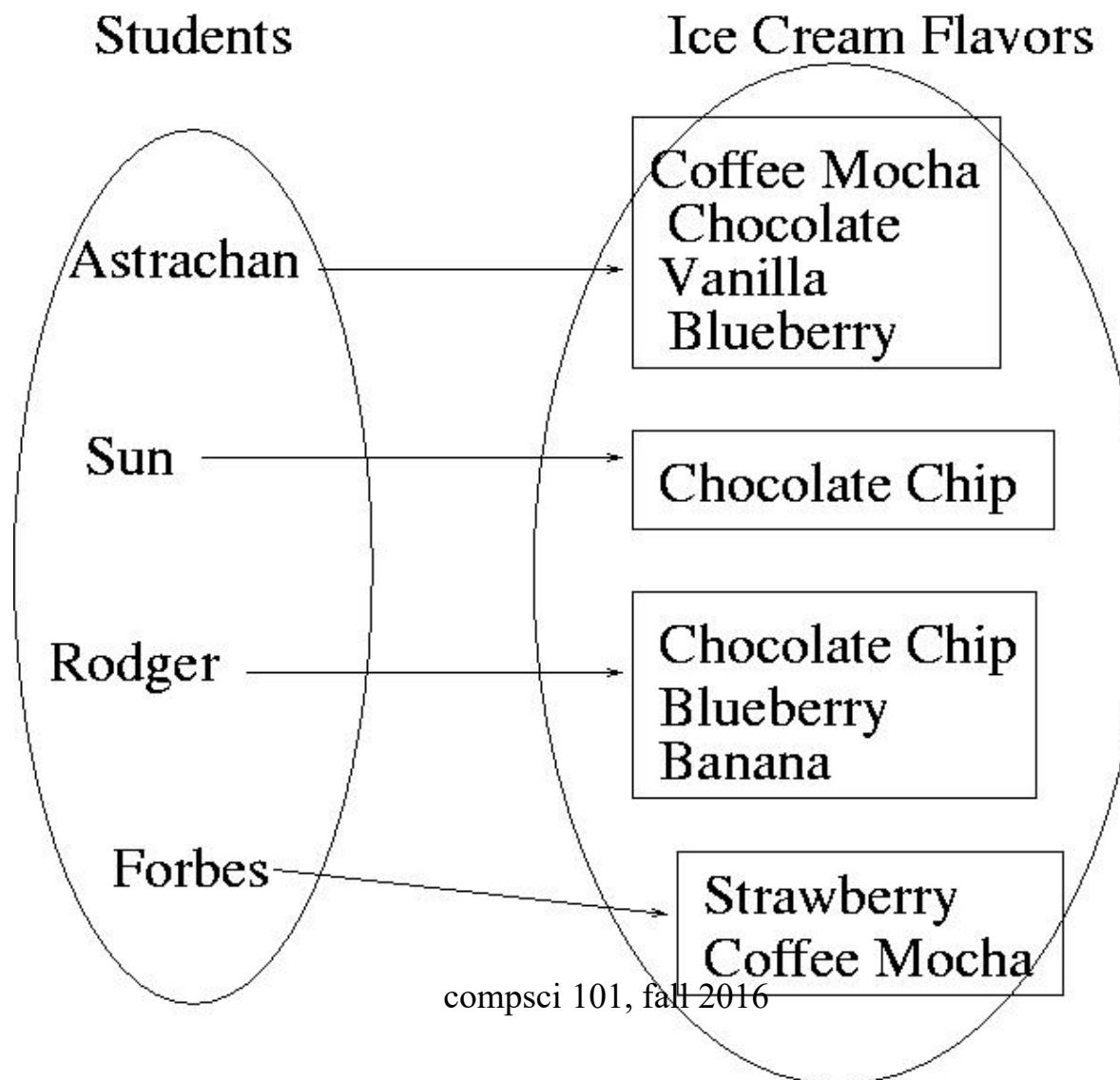


Change Astrachan's value

```
somemap["Astrachan"] = Coffee Mocha
```



Value could be a set or list



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/101f16-1103-2

- Map first names to **count** of corresponding last names

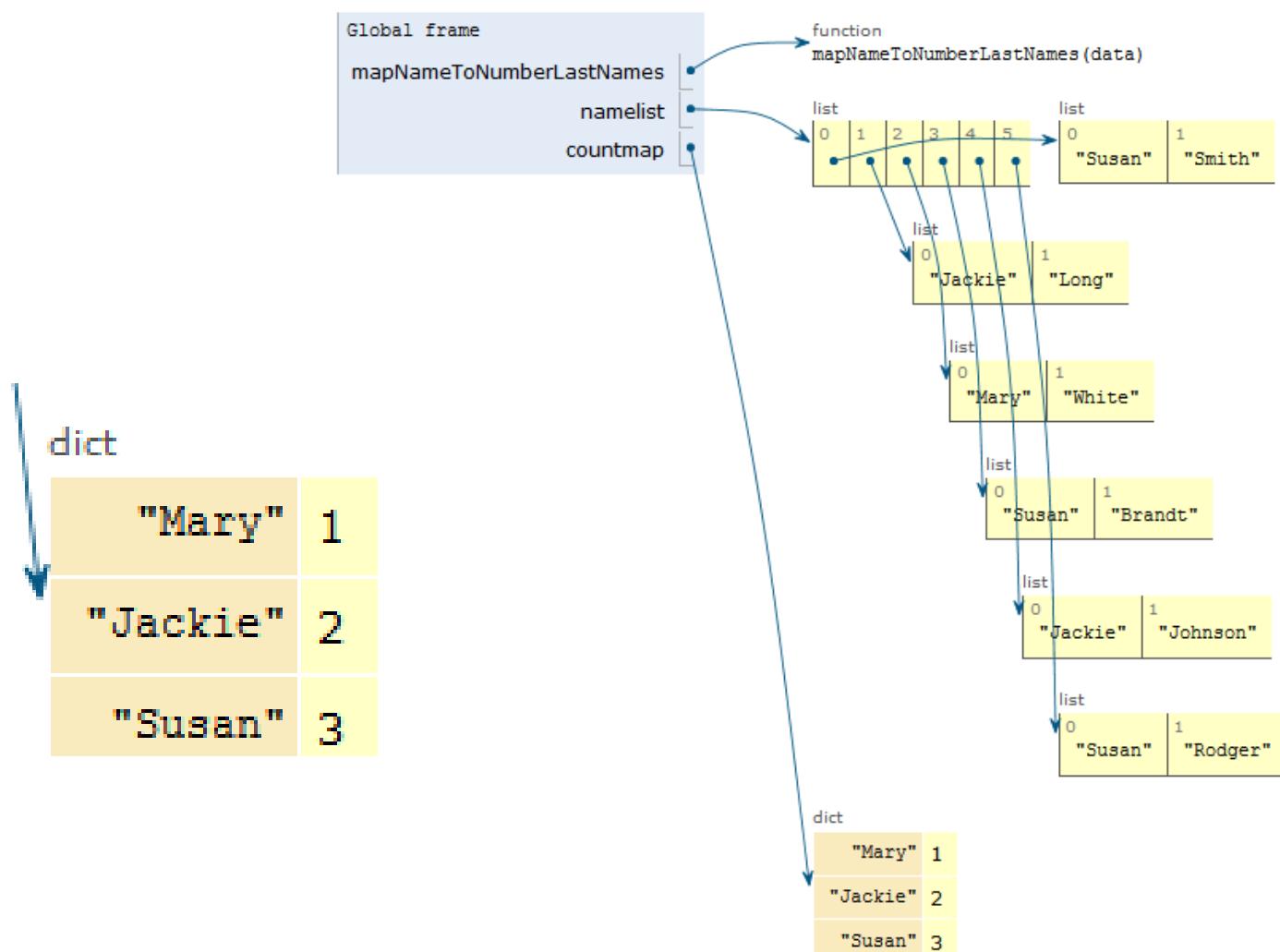
```
def mapNameToNumberLastNames(data):
```

Use a dictionary/map

- popularMap.py

Trace example with Python Tutor

see popularMapSolnSmall.py



Use a dictionary/map

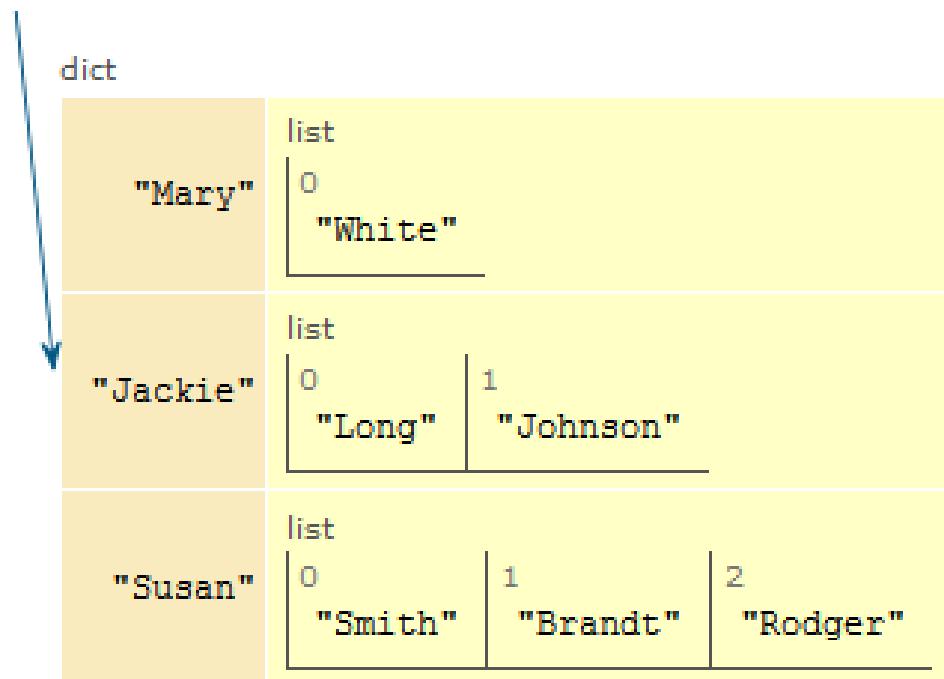
www.bit.ly/101f16-1103-3

- 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 a dictionary/map

www.bit.ly/101f16-1103-4

- 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?