Compsci 101
Problem Solving

Susan Rodger
October 27, 2022
P is for ...

• Python
  • Whatever you want it to be? Language!!!
• Parameter
  • When an argument becomes a variable
• Power Cycle
  • Not the last resort. But works
• P2P
  • From networking to collaboration
The Tech Twins

- Troy and Travis Nunnally
- Between them: 2 master’s and 1 doctorate from Georgia Tech
- Cofounders of Brain Rain Solutions
  - Augmented-reality
  - Internet-of-things
- Applied machine learning

https://www.wired.com/story/what-atlanta-can-teach-tech-about-cultivating-black-talent/

Troy: “My advice would be to stay consistent. Always think persistently and consistently about learning a particular craft.”

Travis: “I think that you have to be passionate and find something that you simply love and enjoy. Not only find that thing — but actually be a lifelong learner around that.”
Announcements

• **Assign 4 GuessWord** due Thursday, Nov 3
  • Sakai Assignment Quiz due **TUESDAY, Nov 1**

• **APT-5 and Assign 5 out, Nov 1 or 2**
  • Will talk about next Thursday

• **Lab 7 Friday**
  • Do prelab 7 before going to lab

• **Prof Rodger moving Thur office hours to 5-6pm permanently!**

• **Exam 2 on Tuesday, Nov 1**
  • See slides from last time
  • Review materials on Nov 1 date on course calendar
PFTD

• Problem Solving and Practice Exam problem
• Jotto game
Problem Solving – What to use

• Do you need to loop over anything?
  • Do you need the index of the item?
• Do you need to make a decision?
• Do you need unique elements?
• Are you working with two groups of things?
  • Are they parallel lists?
  • Are you comparing elements in some way with two groups of elements
• Need an index loop to access two elements in different lists at the same position
• Do you want to put both into sets and use a set operation?
Problem Solving – What to use

• Do you need to loop over anything?
  • Do you need the index of the item?
• Do you need to make a decision?
• Do you need unique elements?
• Are you working with two groups of things?
  • Are they parallel lists?
  • Are you comparing elements in some way with two groups of elements
Now let's look at an APT from APT-5
APT: SandwichBar Search

Problem Statement

It's time to get something to eat and I've come across a sandwich bar. Like most people, I prefer certain types of sandwiches. In fact, I keep a list of the types of sandwiches I like.

The sandwich bar has certain ingredients available. I will list the types of sandwiches I like in order of preference and buy the first sandwich the bar can make for me. In order for the bar to make a sandwich for me, it must include all of the ingredients I desire.

Given available, a list of Strings/ingredients the sandwich bar can use, and a orders, a list of Strings that represent the types of sandwiches I like, in order of preference (most preferred first), return the 0-based index of the sandwich I will buy. Each element of orders represents one type of sandwich I like as a space-separated list of ingredients in the sandwich. If the bar can make no sandwiches I like, return -1.

```python
filename: SandwichBar.py

def whichOrder(available, orders):
    """
    return zero-based index of first sandwich in orders, list of strings that can be made from ingredients in available, list of strings
    """

    # you write code here
```
Sandwich Bar Example

- available = [ "cheese", "cheese", "cheese", "tomato" ]
- orders = [ "ham ham ham", "water", "pork", "bread", "cheese tomato cheese", "beef" ]
WOTO-1 SandwichBar
Possible Exam Question
PROBLEM 3: (Wins and Losses)

Consider the following data file of information on club basketball teams. Each line in the file represents two teams playing each other and their scores. The format of each line in the file is team1, followed by a hyphen, followed by the number of points team1 made, followed by a colon, followed by team2, followed by a hyphen, and followed by the number of points team2 made. The first team on each line is the home team, where the game was played.

An example of the data file is shown below. For example, in the first line, duke was the home team and duke played against unc, with duke scoring 78 points and unc scoring 76 points, so duke won the game.

duke-78:unc-76
unc-87:virginia tech-80
wake forest-73:duke-92
miami-82:unc-79
wake forest-67:miami-77
ncsu-68:unc-70
unc-80:gatech-65
ncsu-77:virginia tech-73
virginia tech-83:wake forest-79
gatech-75:ncsu-81
gatech-81:wake forest-70
duke-76:ncsu-74
virginia tech-75:miami-74
A. Write the function `processinfo` that has one parameter `filename` which represents the name of the file. This function returns a list of lists of items in which each inner list has four items and represents one line from the file. The first item is a string of team1’s name, the second item is the integer number of points team1 scored, the third item is a string of team2’s name, and the fourth item is the integer number of points team2 scored.

For example, the line `data = processinfo("teamdata.txt")` where "teamdata.txt" is the file above would result in `data` having the value on the next page.

duke-78:unc-76
unc-87:virginia tech-80
wake forest-73:duke-92
miami-82:unc-79
wake forest-67:miami-77
ncsu-68:unc-70
unc-80:gatech-65
ncsu-77:virginia tech-73
virginia tech-83:wake forest-79
gatech-75:ncsu-81
gatech-81:wake forest-70
duke-76:ncsu-74
virginia tech-75:miami-74

data = [ ['duke', 78, 'unc', 76], ['unc', 87, 'virginia tech', 80], ['wake forest', 73, 'duke', 92], ['miami', 82, 'unc', 79], ['wake forest', 67, 'miami', 77], ['ncsu', 68, 'unc', 70], ['unc', 80, 'gatech', 65], ['ncsu', 77, 'virginia tech', 73], ['virginia tech', 83, 'wake forest', 79], ['gatech', 75, 'ncsu', 81], ['gatech', 81, 'wake forest', 70], ['duke', 76, 'ncsu', 74], ['virginia tech', 75, 'miami', 74] ]
Complete the function processinfo below.

```python
def processinfo(filename):
    f = open(filename)
```
B. Write the function `schoolsBeat` that has two parameters, `data` and `team`, where `data` is the list of lists in the format from Part A, and `team` is a string. This function returns a list of tuples, where each tuple is information about a game that `team` won. Each tuple has the name of the team beat, followed by the number of points they won by.

For example, assume `data` is the lists of lists of four items on the previous page. The two examples below show the result of calling `schoolsBeat` with this filename and a team name. For example, duke beat three teams, ncsu by 2 points, unc by 2 points and wake forest by 19 points, wake forest did not beat any teams, and unc beat three teams.

<table>
<thead>
<tr>
<th>call</th>
<th>returns</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>schoolsBeat(data, ”duke”)</code></td>
<td><code>[('ncsu', 2), ('unc', 2), ('wake forest', 19)]</code></td>
</tr>
<tr>
<td><code>schoolsBeat(data, ”wake forest”)</code></td>
<td><code>[()]</code></td>
</tr>
<tr>
<td><code>schoolsBeat(data, ”unc”)</code></td>
<td><code>[('gatech', 15), ('ncsu', 2), ('virginia tech', 7)]</code></td>
</tr>
</tbody>
</table>

def schoolsBeat(data, team):

data = [ ['duke', 78, 'unc', 76],
         ['unc', 87, 'virginia tech', 80],
         ['wake forest', 73, 'duke', 92],
         ['miami', 82, 'unc', 79], ...]
Jotto: Game similar to GuessWord

- [http://jotto.augiehill.com/single.jsp](http://jotto.augiehill.com/single.jsp)
- No letters repeat – have to agree on this
- Shall we play a game?
Write program where Computer Guesses Your Word

• You give the computer a word to guess, called wordToGuess

• Computer does brute force, no thinking or eliminating letters
  • It picks a word at random
  • Calculates how many letters in common with wordToGuess, say x letters
  • Only keep words with x letters in common
  • Repeats until guesses the word
We will build useful functions to use to build the game
WOTO-2 Approaching Implementation


- What is needed?
- What order should the code do things?
WOTO-3 More on Jotto

• What is needed?
• What order should the code do things?