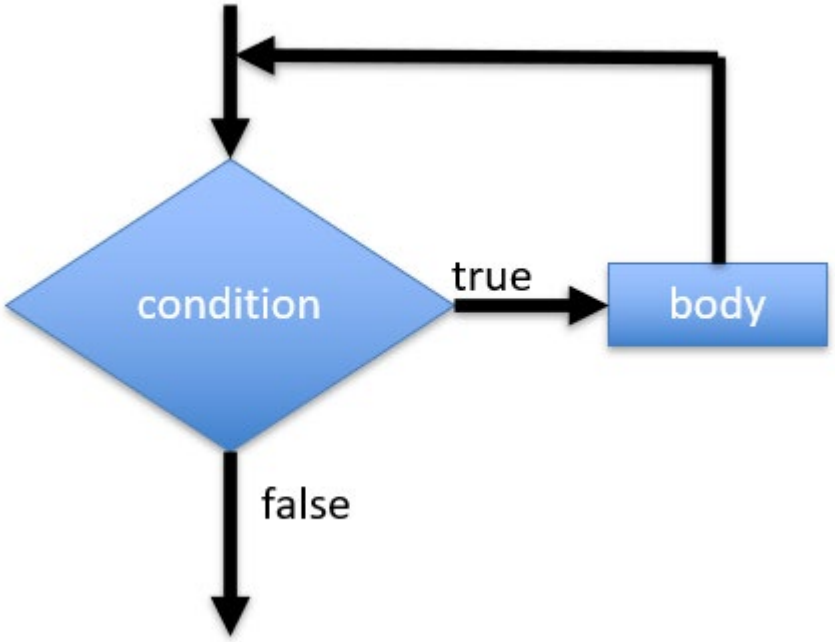
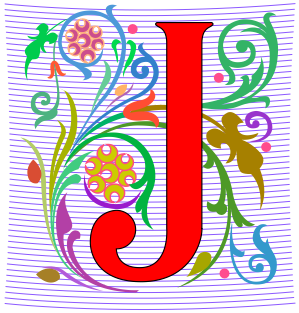
Compsci 101
Files, While loops, Bagels



Susan Rodger

February 10, 2022

J is for …



- **JSON**
  - Format for data transmitted across the web
- **JPEG**
  - Image format based on lossy compression
- **Jacquard Loom**
  - 1804 "automated" loom

Latanya Sweeney

PhD. Computer Science, MIT – first black woman

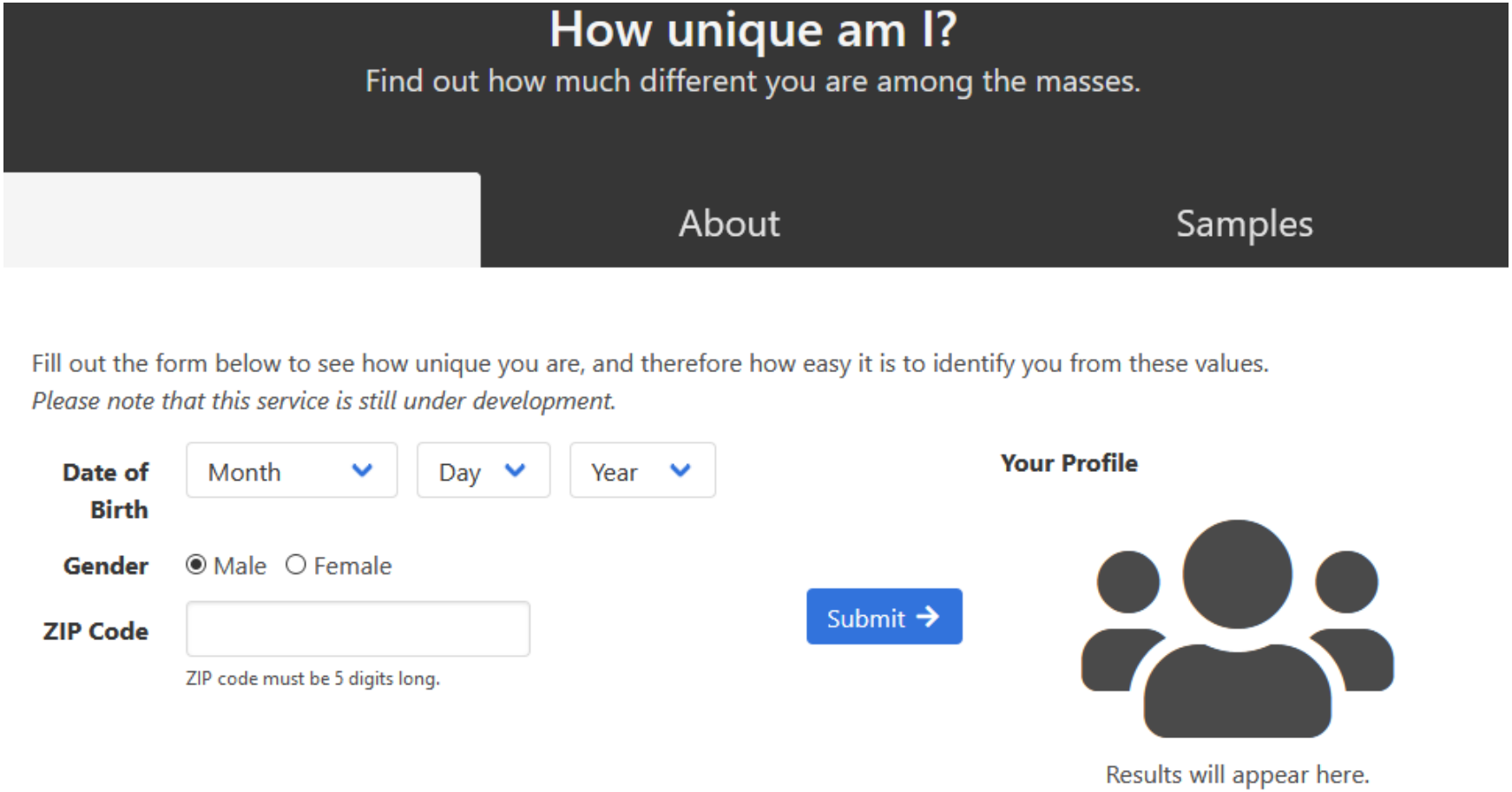
Over 100 publications, Fellow ACMI

"I am a computer scientist with a long history of weaving technology and policy together to remove stakeholder barriers to technology adoption. My focus is on "computational policy" and I term myself a "computer (cross) policy" scientist. I have enjoyed success at creating technology that weaves with policy to resolve real-world technology-privacy clashes.

http://latanyasweeney.org/

Identify 87% of US population using (dob,zip,gender). Prof. Government and Technology @ Harvard, instrumental in HIPAA because if *de-identification* work. Former CTO of the Federal Trade Comm.

One of her websites you can try: https://aboutmyinfo.org/identity



Your Profile

Gender: Female

ZIP Code: [redacted] (pop. 46282)

Date of Birth	[redacted]	Easily identifiable by birthdate (about 1).
Birth Year	[redacted]	Lots with your birth year (about 273).
Range	[redacted] to [redacted]	Wow! There are lots of people in the same age range as you (about 1365).

Five year range

Announcements

- APT-2 due tonight!
- APT-3 out, due in one week
- Assignment 2 due Tuesday, Feb. 15

- Lab 5 Friday – Do Prelab before going

- APT Quiz 1 coming … 2/24-2/27
  - APTs you take some time during this period
  - Take online
  - Two APT Quizzes – 2cd one 4/7-4/10

PFTD

- Files and Data
- While loops and Collatz sequence
- Bagel APT

Text File Processing Pattern

- See module **`FileStuff.py`**
  - If newline **`'\n'`** is read, call **`.strip()`**
  - If want to break line into "words", call **`.split()`**

- Process the list that is returned by **`.split()`**
  - May need to convert strings to int or float or …

- The **`for line in f:`** pattern is efficient
  - Contrast list returned by **`f.readlines()`**

FileStuff.py: avgWord

```
def avgWord(fname):
    f = open(fname,encoding="utf-8")
    totalWords = 0
    totalLen = 0
    for line in f:
        line = line.strip()  #remove newline
        data = line.split()
        for word in data:
            totalWords = totalWords + 1
            totalLen = totalLen + len(word)

    f.close()
    return totalLen/totalWords
```

Run FileStuff

```
20  if __name__ == '__main__':
21      files = ["poe.txt", "confucius.txt", "kjv10.txt","oz.txt","species.txt"]
22      for f in files:
23          avg = avgWord("data/"+f)
24          print(f,avg)
```

Output:

```
poe.txt 4.601549053356282
confucius.txt 4.398126192817072
kjv10.txt 4.245566037162798
oz.txt 4.496446700507614
species.txt 5.036
```

Files - Summary

- Open file: f = open(filename)
- "Process" file (2 different ways):
  - for line in f: # get one line at a time with "\n"
  - x = f.readlines() # x is a list of lines with "\n"
- Close file: f.close()

- To think about when processing lines
  - Line is a string with "\n" – .strip() it
  - Maybe .split() line into list of strings (words)?
  - Convert string to int or float - int("376")

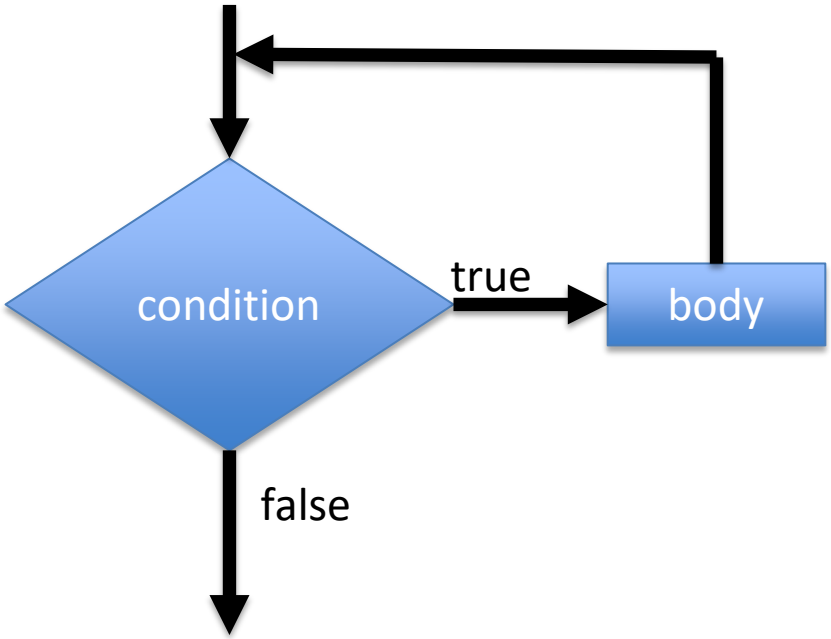
WOTO-1 Files
http://bit.ly/101s22-0210-1

When is a game of chess over?

- If you were to write a program to play chess
  - how many rounds in a game?

Another type of loop: While loop

- Repetition when you stop a loop based on a condition
- while CONDITION:
  BODY



- As long as condition is true, keep executing loop body.
- Must have an update in the body to get closer to condition being false

Example: while

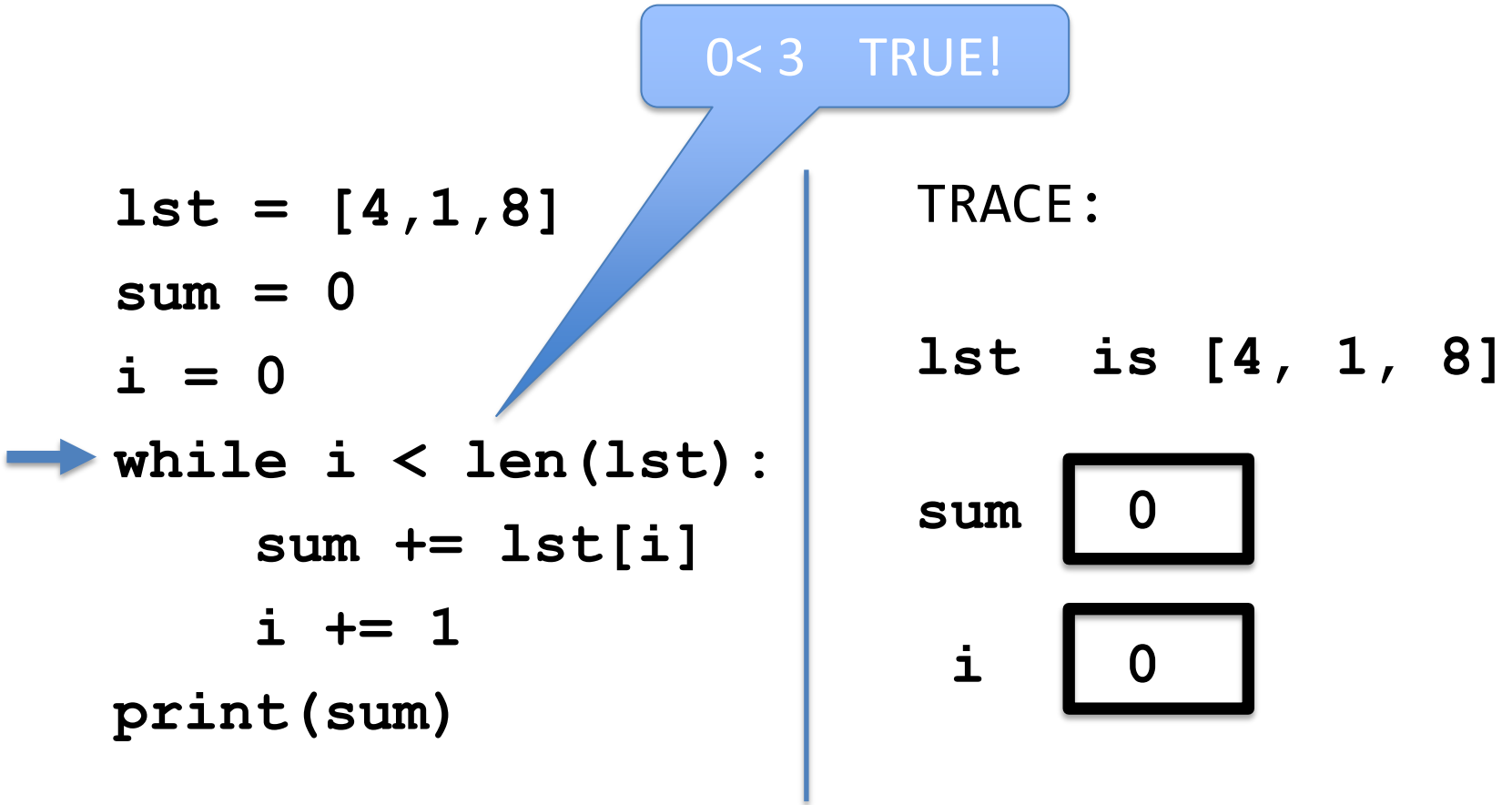
- Playing chess

```
while (game not over)
        make a move in the game
        (game must get closer to ending)
```

Example: while loop – sum list

```
lst = [4,1,8]
sum = 0
i = 0
while i < len(lst):
    sum += lst[i]
    i += 1
print(sum)
```

Example: while loop – sum list



Summary: while loop

```
lst = [4,1,8]
sum = 0
i = 0
while i < len(lst):
    sum += lst[i]
    i += 1
print(sum)
```

Initialize loop variable

Check condition, True or false?

update loop variable, should make condition closer to being false

History: From while to for loops

while loop (sum list)

```
lst = [4,1,8]
sum = 0
i = 0
while i < len(lst):
    sum += lst[i]
    i += 1
print(sum)
```

for loop (sum list)

```
lst = [4,1,8]
sum = 0
for n in lst:
    sum += n
print(sum)
```

Alternative while -while True

initialize

```
while True:
    if something:
        break
    if something2:
        update
    update
Continue or return
```

while condition vs while True

while *condition*:

    body

continue

while True:

    body

    if condition:

        break

    continue

While condition is true - must update

- must get closer to making condition false
- use break to exit

Compare: while - while True

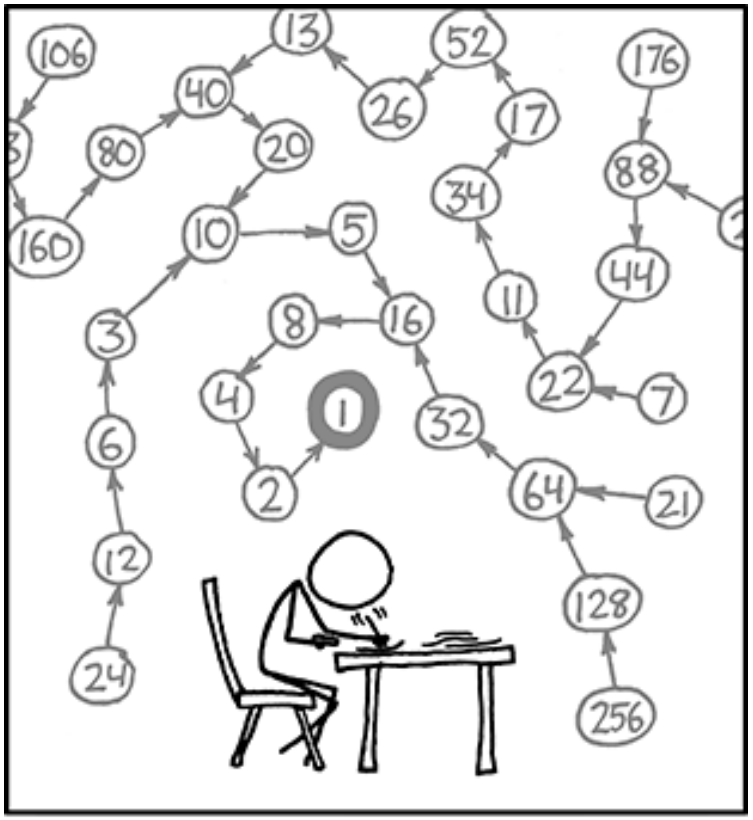
```
lst = [4,1,8]
sum = 0
i = 0
while i < len(lst):
    sum += lst[i]
    i += 1
print(sum)
```

```
lst = [4,1,8]
sum = 0
i = 0
while True:
    if i >= len(lst):
        break
    sum += lst[i]
    i += 1
print(sum)
```

WOTO-2 While loops
http://bit.ly/101s22-0210-2

Now let's see a problem that needs a while loop

https://xkcd.com/710/



THE COLLATZ CONJECTURE STATES THAT IF YOU PICK A NUMBER, AND IF IT'S EVEN DIVIDE IT BY TWO AND IF IT'S ODD MULTIPLY IT BY THREE AND ADD ONE, AND YOU REPEAT THIS PROCEDURE LONG ENOUGH, EVENTUALLY YOUR FRIENDS WILL STOP CALLING TO SEE IF YOU WANT TO HANG OUT.

Collatz Conjecture (Hailstone)

If number is even
    Divide by 2
If number is odd
    multiply by 3 and add 1

Always end up at 1!

Why Solve This? In Python?

- https://en.wikipedia.org/wiki/Collatz_conjecture
- We want to illustrate an indefinite loop
  - One of many mathematical sequences, but …
- There's an XKCD comic about it!
  - Not everyone enjoys XKCD, but …
- Mathematics is foundational in computer science, but
  - Not everyone enjoys logic/math puzzles, but …

Developing and Reasoning about While Loops

- Don't know: *how many times* loop executes
  - *a priori* knowledge, we'll know afterword

- Do know: condition that should be true after loop
  - Its negation is the expression for `BOOL_CONDITION` (loop guard)

```
while BOOL_CONDITION:
    LOOP_BODY
    # modify variables, affect expression
```

Concrete Example: Collatz/Hailstone

- Don't know: *how many times* loop executes
  - some numbers: long sequences, others short

- Do know: condition that should be true after loop
  - It's negation is the expression for loop guard!
  - What is true after loop below finishes?

```
while value != 1:
    loop body
    # modify value somehow
```

Collatz Code

```
6   def hailstone(start, printing=False):
7       """..."""
14      steps = 0
15      current = start
16      while current != 1:
17          if printing:
18              print("{:3d}\t{:6d}".format(steps,current))
19          if current % 2 == 0:
20              current //= 2
21          else:
22              current = current * 3 + 1
23          steps += 1
24
25      if printing:
26          print("{:3d}\t{:6d}".format(steps,current))
27      return steps
```

Sample run

```
44 ▶ if __name__ == '__main__':
45       num = 6
46       s = hailstone(num, True)
47       print('num =',num,'steps =',s)
```

Output:

```
0       6
1       3
2      10
3       5
4      16
5       8
6       4
7       2
8       1
num = 6 steps = 8
```

Collatz Data – Average no. of steps

- How do we gather data for numbers <= 10,000?
  - In general for numbers in range(low,high) ?
  - Call function, store result, store 10,000 results?

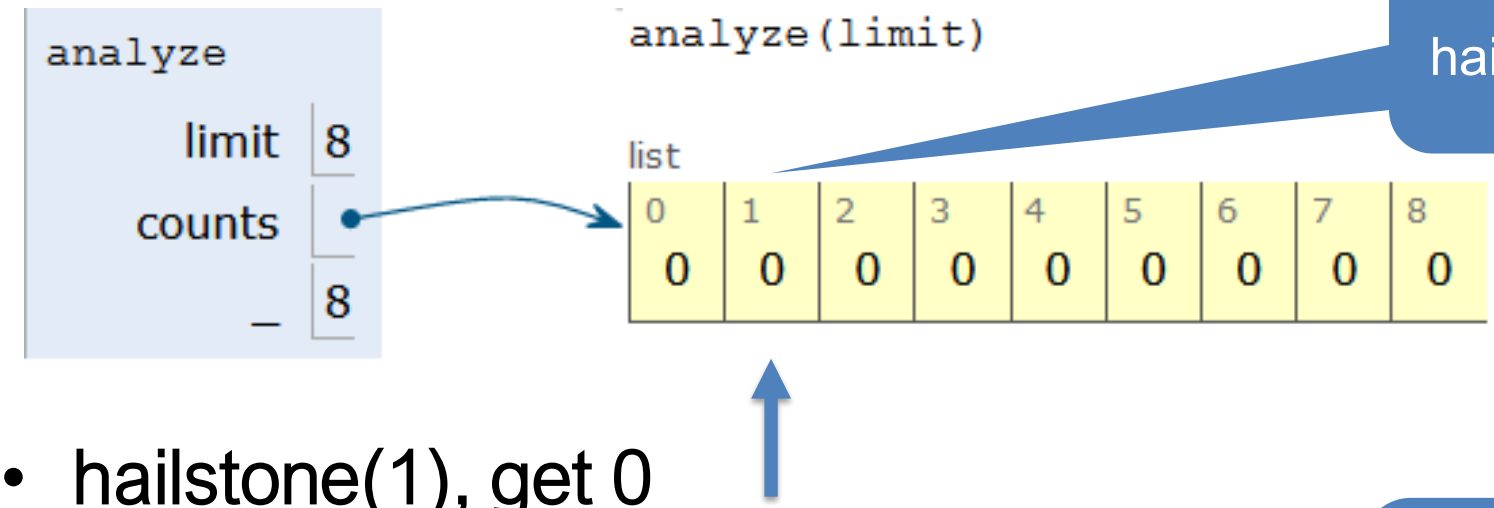
- We'd like counts[k] to be length of sequence for k
  - How do we allocate 10,000 list elements?
  - Like there is `"hello" * 3`
  - There is `[0] * 10000`

Analysis in Collatz.py

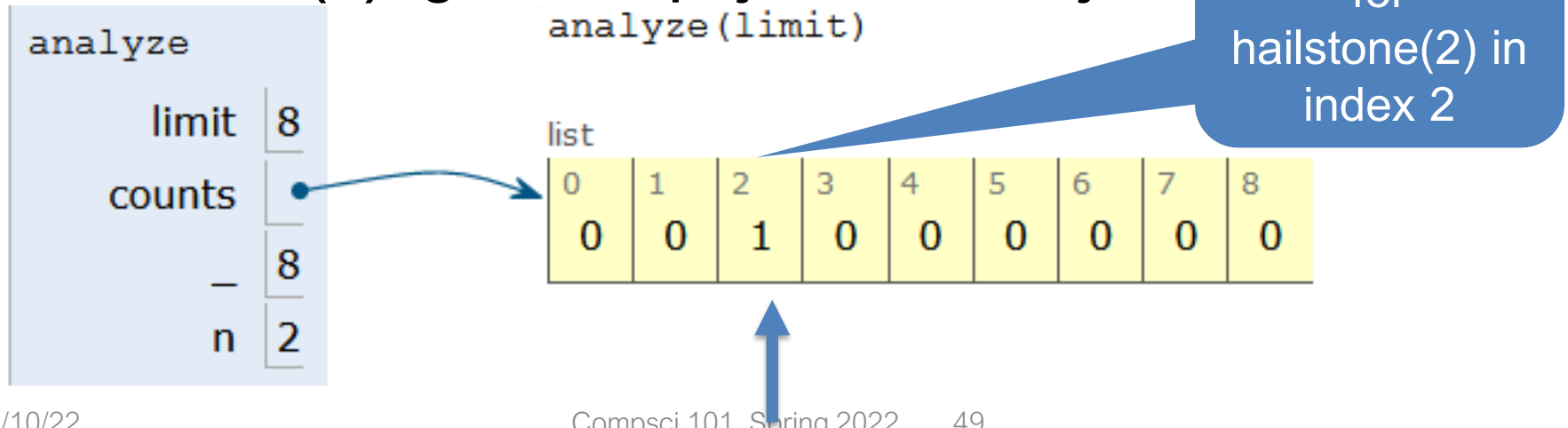
```
29  def analyze(limit):
30      counts = []
31      # max index into count is limit, but start at 1
32      for _ in range(limit+1):
33          counts.append(0)
34
35      for n in range(1, limit+1):
36          counts[n] = hailstone(n)
37
38      avg = sum(counts)/len(counts)-1  # ignore index 0
39      mx = max(counts)
40      dex = counts.index(mx)
41      print("average",avg)
42      print("max is %d at %d" % (mx,dex))
```

counts list when limit is 8?

- Counts is of size 8+1, we ignore slot 0



- hailstone(1), get 0
- hailstone(2), get 1 step, just divide by 2

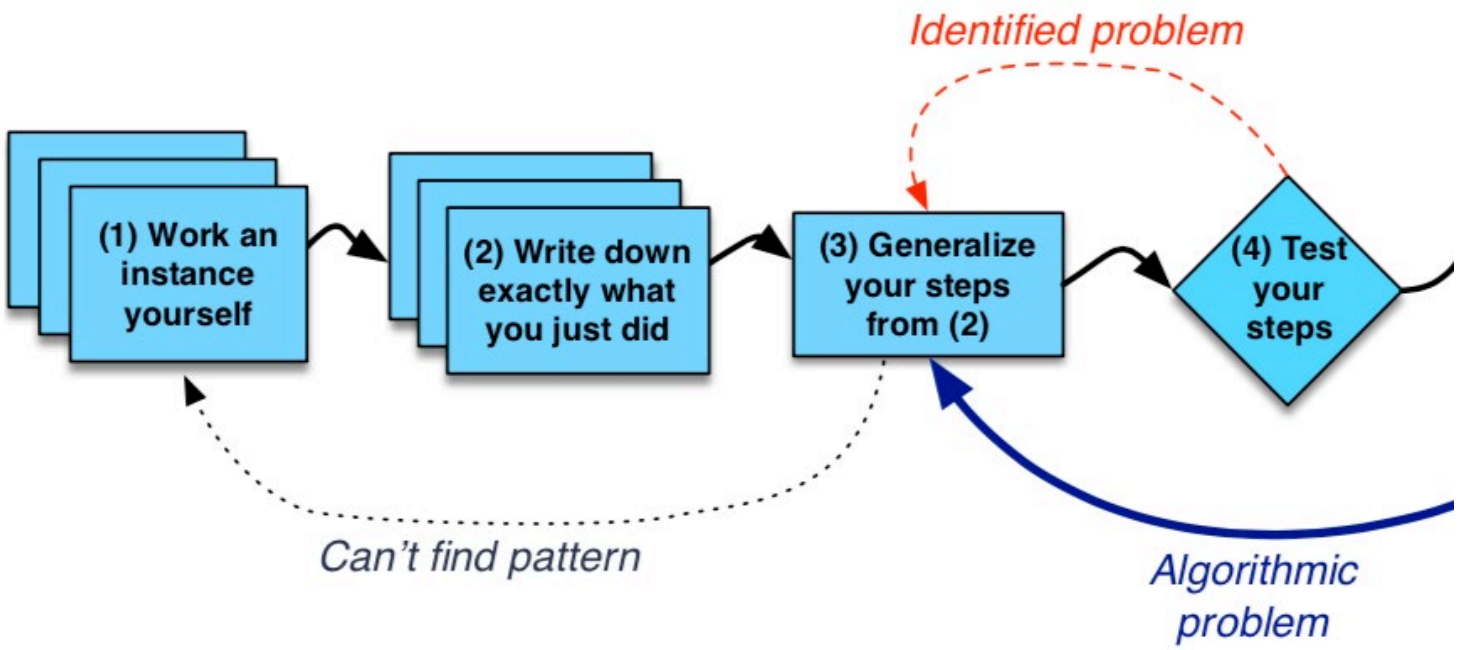


WOTO-3 Collatz and While
http://bit.ly/101s22-0210-3

Bagels (Accumulation)

APT Bagels

- How figure out how many bagels needed?
  - 7-steps!



APT: Bagel Counting

Problem Statement

You are in charge of web-based orders for your neighborhood bagel store, *The Bagel Byte*. Each evening you must total the orders to be picked up the next day. Some orders are simply for *N* bagels, but each order of a dozen or more bagels is topped off with an extra bagel, the so-called "baker's dozen". This means, for example, that an order for 25 bagels actually requires 27 bagels to fulfill since there are two extra bagels needed for each dozen in the order. An order for 11 bagels doesn't require any extra since it's for less than a dozen.

Class

```
filename:  Bagels.py

def bagelCount(orders) :
    """
    return number of bagels needed to fulfill
    the orders in integer list parameter orders
    """

    # you write code here
```

Given a list of integers representing bagel orders determine the number of bagels needed to fulfill all the orders.

Examples

Examples

1.
```
orders = [1,3,5,7]

Returns:  16
```

No order is for more than a dozen, return the total of all orders.

2.
```
orders = [11,22,33,44,55]

Returns: 175 since 11 + (22+1) +(33+2) + (44+3) + (55+4) = 175
```

Step 1 and 2

- Step 1: Solve an instance (think)
  - orders = [11, 3, 24, 17]

WOTO-4 Step 3: Generalize
http://bit.ly/101s22-0210-4