

## Compsci 6/101: Re[gex|cursion]

- Regular expressions and grammars
  - Recognizing and validating text based on rules
  - Generating text based on rules
  - From programs to scholarly papers
- Recursion: self-referential structures and code
  - Look up recursion in [Google | Bing]
  - Look-it up in the index of ...
  - What is a folder on your computer's desktop?

Compsci 06/101, Spring 2011

16.1

## What's the deal with self-reference?

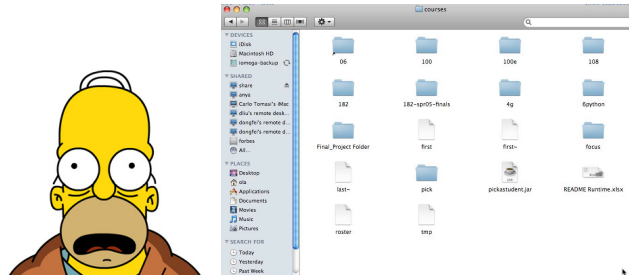
- This sentence is false.
  - This sentence is true.
  - I think I'm thinking too much.
- Recursive functions
  - Function doesn't call itself, each function is a separate "thing", with its own state
  - Must have a *base case*, no recursive calls made, no self-referential work done
  - Each recursive call gets closer to base case
    - Using some metric that we can reason about

Compsci 06/101, Spring 2011

16.2

## What's in a folder on your computer?

- Where are the large files? How do you find them?
  - Can a folder be inside a folder? Why?



Compsci 06/101, Spring 2011

16.3

## Finding large files: FileVisit.py

```
def bigfiles(dirname,min_size):
    large = []
    #print dirname
    for sub in os.listdir(dirname):
        path = os.path.join(dirname,sub)
        if os.path.isdir(path):
            large.extend(bigfiles(path,min_size))
        else:
            size = os.path.getsize(path)
            if size > min_size:
                large.append((path,size))
    return large

bigfs = bigfiles("c:\\Users",10000)
[(file,102030),(nfile,1030303),(pfile,10001)]
```

Compsci 06/101, Spring 2011

16.4

## Dissecting FileVisit.py

- How do we find the contents of a folder?
  - Another name for folder: directory
  - How do we identify folder? (by name)
  - `os.listdir(dirname)` returns a list of ...
  - Path is `c:\user\ola\foo` or `/Users/ola/bar`
  - `os.path.join(dir,sub)` returns full path
  - Platform independent paths
- What's the difference between file and folder?
  - `os.path.isdir()` and `os.path.getsize()`

CompSci 06/101, Spring 2011

16.5

## Creativity with self-reference

- Sometimes madlibs are fun (corollary?)
    - Humans fill in the blanks
    - Computers automatically fill in the blanks
- The <apt-name> APT was really <description> but I didn't do it because I <excuse>
- <description> :: "cool", "terrible", "baller", ...  
<excuse> :: "was too tired", "didn't know how", ...  
<excuse> :: <excuse> and <excuse>
- See SimpleGrammar.py

CompSci 06/101, Spring 2011

16.6

## Why look at recursion?

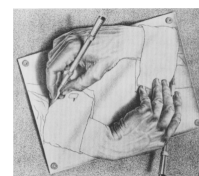
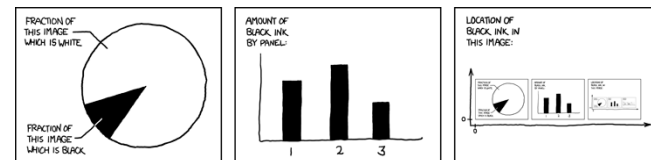
- Powerful tool for elegantly expressing algorithms
  - Never necessary, but alternative can be hard to develop, lengthy, tricky, ... (but then again ...)
  - Part of essential toolkit of computer scientist
    - Arguably not essential for web developer, entrepreneur, social media promoter, ...
- Another way of thinking about problems
  - In every day life? When writing code?
  - When drawing? When creating symphonies?

CompSci 06/101, Spring 2011

16.7

## Recursion in Pictures

- <http://xkcd.com/688/> and <http://xkcd.com/543/>



CompSci 06/101, Spring 2011

16.8

## New sorting algorithms happen ...

- **timsort is standard on...**
  - Python as of version 2.3, Android, Java 7
  - According to <http://en.wikipedia.org/wiki/Timsort>
    - Adaptive, stable, natural mergesort with supernatural performance
- **What is mergesort? Fast and Stable: see Merge.py**
  - What does this mean?
  - Divide and Conquer, what does that mean?
  - Nothing is faster, what does that mean?
  - Quicksort is faster, what does that mean?

## How to merge/join two sorted lists?

- **['a','d','z'] and ['b','c','f','y']**
    - **['a','b','c','d','f','y','z']**
- ```
def merge(left, right):
    result, i, j = [], 0, 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result.extend(left[i:])
    result.extend(right[j:])
    return result
```

## How to sort a list?

```
def mergesort(list):
    if len(list) < 2:
        return list
    else:
        middle = len(list) / 2
        left = mergesort(list[:middle])
        right = mergesort(list[middle:])
        return merge(left, right)
```

## Comparing Algorithms

- **Searching a list of N elements for a specific value**
  - Worst case is ....
- **Doing binary search (guess a number), sorted list**
  - Worst case is ...
- **Sorting using merge sort**
  - Worst case is ...
- **Naïve finding which ball collides with specific ball**
  - Worst case is ...



## Brian Reid (Hopper Award 1982)

Feah. I love bubble sort, and I grow weary of people who have nothing better to do than to preach about it. Universities are good places to keep such people, so that they don't scare the general public.

(continued)



## Brian Reid (Hopper 1982)

I am quite capable of squaring  $N$  with or without a calculator, and I know how long my sorts will bubble. I can type every form of bubble sort into a text editor from memory. If I am writing some quick code and I need a sort quick, as opposed to a quick sort, I just type in the bubble sort as if it were a statement. I'm done with it before I could look up the data type of the third argument to the quicksort library.

I have a dual-processor 1.2 GHz Powermac and it sneers at your  $N$  squared for most interesting values of  $N$ . And my source code is smaller than yours.

Brian Reid  
who keeps all of his bubbles sorted anyhow.



## Niklaus Wirth (Turing award 1984)

I have read your article and share your view that Bubble Sort has hardly any merits. I think that it is so often mentioned, because it illustrates quite well the principle of sorting by exchanging.

I think BS is popular, because it fits well into a systematic development of sorting algorithms. But it plays no role in actual applications. Quite in contrast to C, also without merit (and its derivative Java), among programming codes.



## Owen O' Malley

- Debugging can be frustrating, but very rewarding when you find the cause of the problem. One of the nastiest bugs that I've found when I was at NASA and the Mars Explorer Rovers had just landed on Mars.

<http://sortbenchmark.org/>

Hadoop sets Terabyte sort record

- Java
- 34252 nodes
- Greedy!

