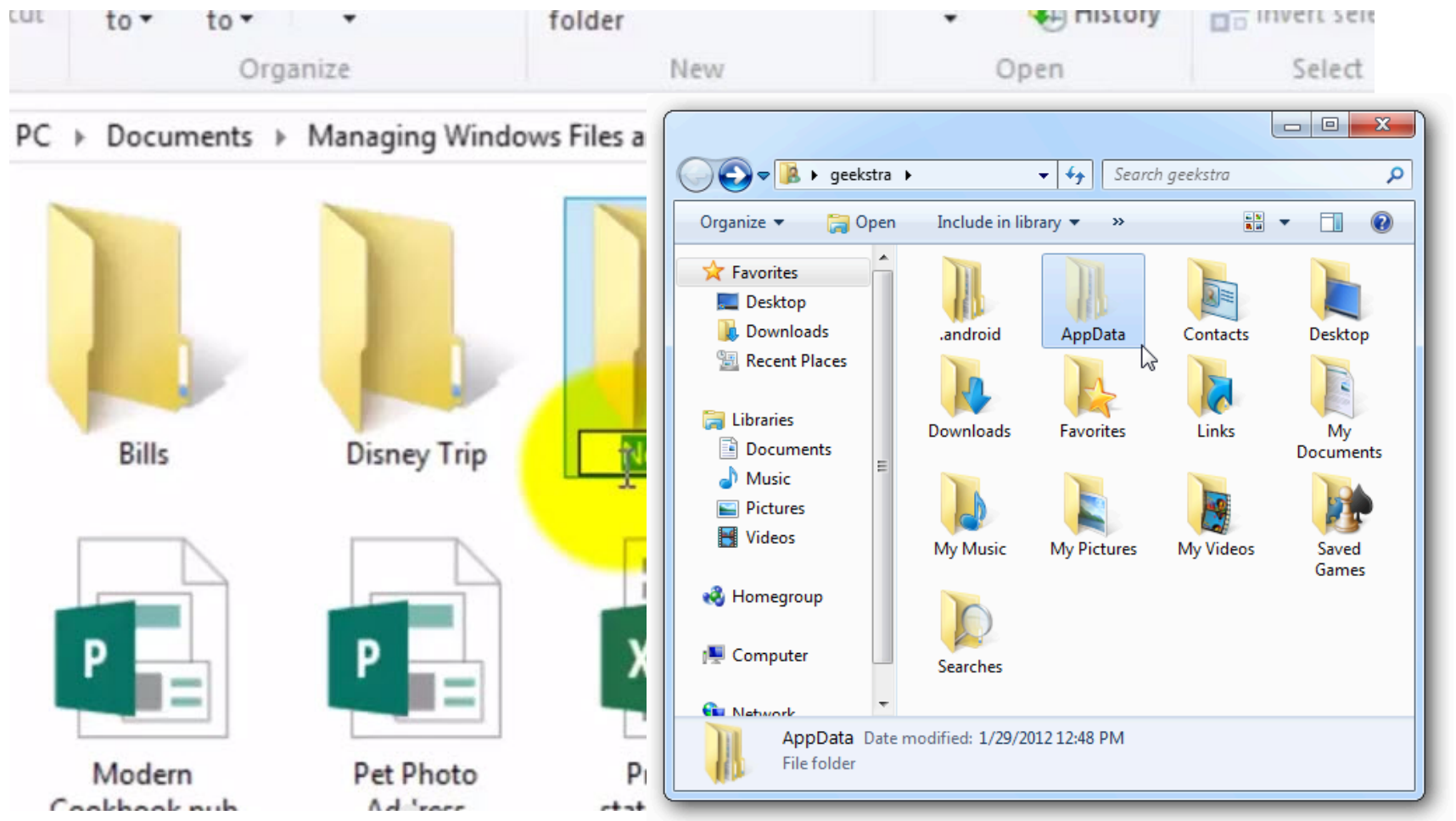


Plan for ThBThB

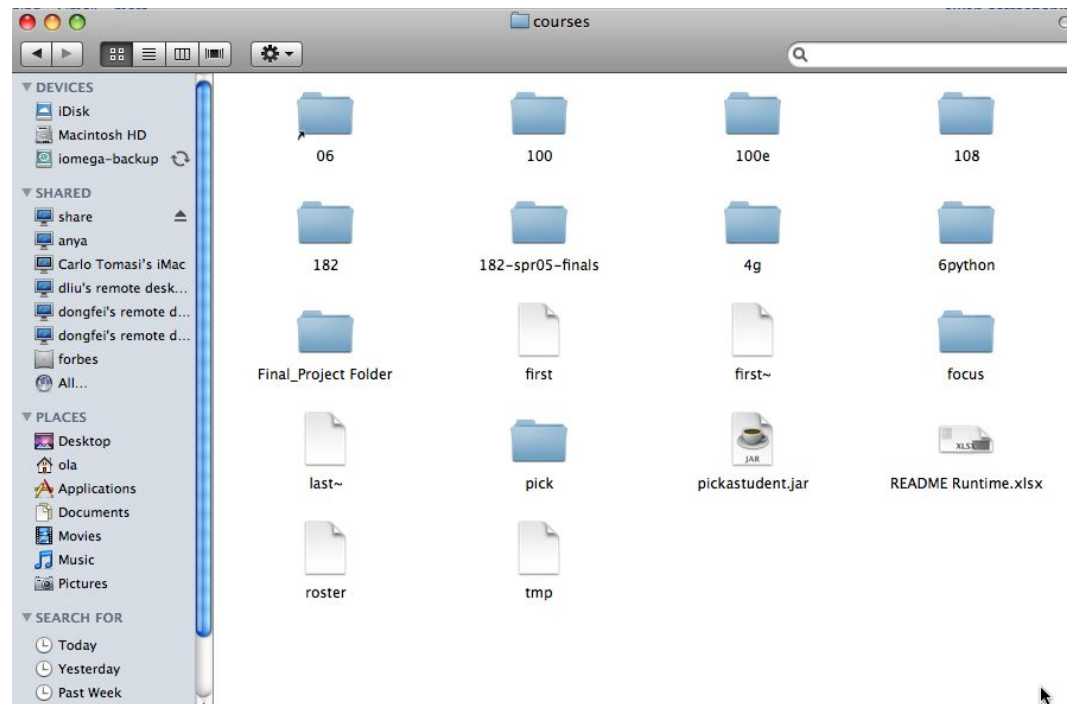
- **Programming and understanding ...**
 - **Hierarchical structures and concepts**
 - What is a file system on a computer?
 - What is the Internet?
 - How does the Domain Name System Work?
- **How do you access directories?**
 - And all the files in a directory, and the ...
- **How do you get recommendations**
 - Based on Yelp, Amazon, Netflix,

What's in a file-system Folder?



What's in a folder on your computer?

- Where are the large files? How do you find them? They take up space! Erase? Backup?
 - Can a folder be inside a folder? Why?



Finding large files: FileVisit.py

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

biggs = bigfiles("/Users/ola/Documents",10000)
```

Does the function call itself? No!

```
def visit(dirname):  
    for inner in dirname:  
        if isdir(inner):  
            visit(inner)  
        else:  
            print name(inner), size(inner)
```

- Is a file inside itself? No!
- Does pseudo code make sense?
 - Details make this a little harder in Python, but close!

Structure matches Code

- **If you see a folder**

- Find large files, if you find a folder, ...
- To compress folder, must compress ...
 - Files and repeat process for folders

- **Structure of list of lists**

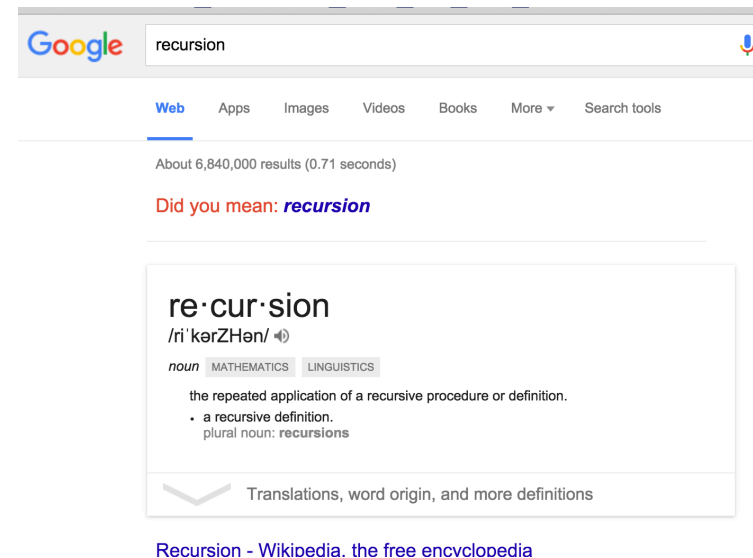
- Can also lead to processing a list which requires processing a list which ...

[[[a,b], [c,d], [a, [b,c],d]]

(a * (b + c (d + e*f)) + (a* (b+d)))

Recursion

- What is recursion? What if we ask Google?
- Structure matches code, or concept matches code
 - Must have a base case when no recursive call made
- Simpler/smaller calls
- The case of Quicksort
 - Inventor didn't get it



Sir Anthony (Tony) Hoare

There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. And the other way is to make it so complicated that there are no obvious deficiencies.



Turing Award,
apologizes for null
reference: \$\$\$ mistake

The os and os.path libraries

- Use an API to isolate system dependencies
 - C:\\x\\y
 - /Users/ola/Desktop
- FAT-32, ReFS, WinFS, HFS, HSF+, fs
 - Underneath, these systems are different
 - Python API insulates and protects programmer
- Why do we have `os.path.join(x, y)`?

Finding large files: FileVisit.py

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

biggs = bigfiles("/Users/ola/Documents",10000)
```

Dissecting FileVisit.py

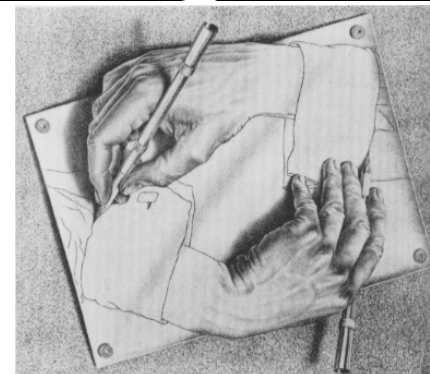
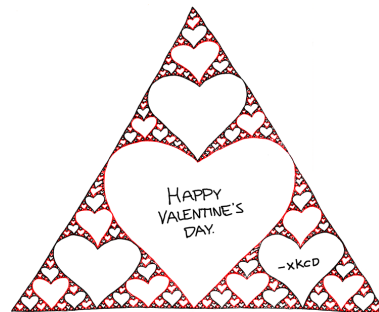
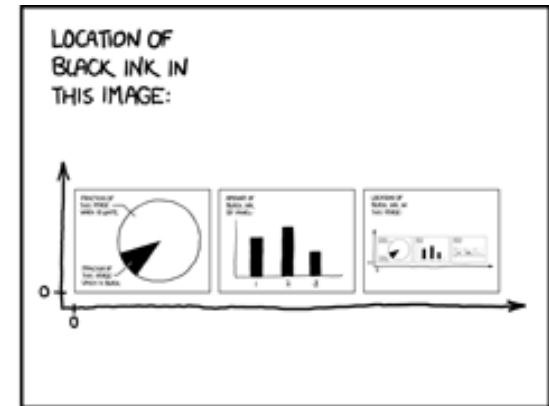
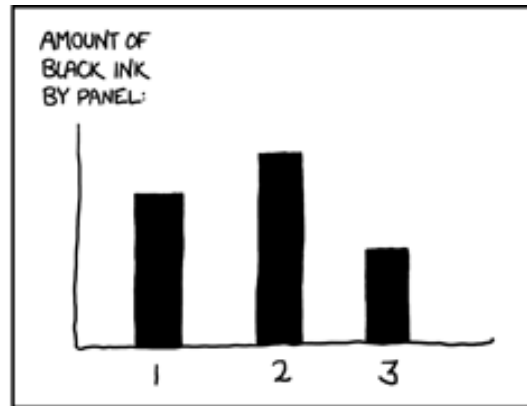
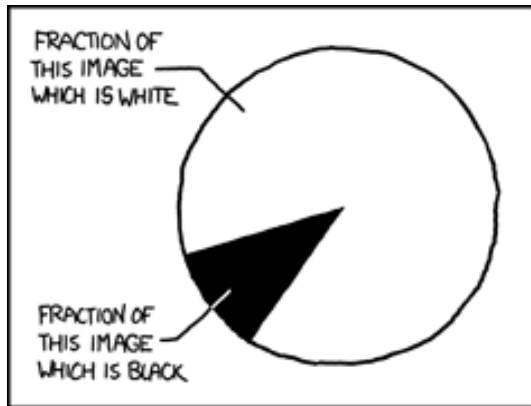
- **How do we find the contents of a folder?**
 - Another name for folder: directory
- **How do we identify folder? (by name)**
 - `os.listdir(dirname)` returns a list of files and folder
- **Path is `c:\user\ola\foo` or `/Users/ola/bar`**
 - `os.path.join(dir,sub)` returns full path
 - Platform independent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

os. And os.path questions

<http://bit.ly/101fall15-nov19-1>

Recursion in Pictures

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



The power of working collaboratively

- **Interdisciplinary:**
 - **Music and Compsci**
(for Compsci 308 final project)

The final product is so much more than we had hoped for though it was something that we aimed for from the beginning.

Our investment into a huge and meticulous design process was a huge factor in making later progress. 35000+ lines of code / design / documentation gave us a project we were all very happy and proud to be a part of.

Compsci 101.2, Fall 2015

- **Who is Ge Wang?**

<http://www.youtube.com/watch?v=ADEHmkL3HBg>



Badges APT from Quiz3

- <http://www.cs.duke.edu/csed/pythonapt/badges.html>
- Use abstraction think about structure
 - Similar to os.path?



```
def isContainedIn(seta, setb) :
    return True
def findLabel(labels, deeds, needs) :
    ds = set(deeds)
    for i in range(len(needs)) :
        me = needs[i].split()
        setme = set(me)
        if isContainedIn(setme, ds) :
            return labels[i]

    return "nobadge"
```

Math, Engineering, Sociology

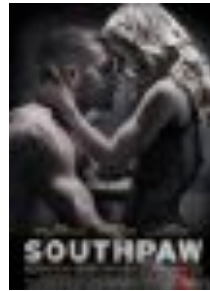
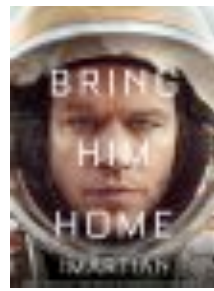
- Netflix prize in 2009
 - Beat the system, win
 - <http://nyti.ms/sPvR>



Collaborative Filtering

- **How does Amazon know what I want?**
 - Lots of customers, lots of purchases
- **How does Pandora know music like Kanye's?**
 - This isn't really collaborative filtering, more content-based
- **How doe Netflix recommend movies?**
 - Why did they offer one million \$\$ to better their method?
- **Students at Duke who like Compsci also like ...**
 - Could this system be built?

From User Rating to Recommendations



Spectre	Martian	Southpaw	Everest	PitchPerfect 2
3	-3	5	-2	-3
2	2	3	2	3
4	4	-2	1	-1

- **What should I choose to see?**
 - What does this depend on?
- **Who is most like me?**
 - How do we figure this out