CompSci 101
Introduction to Computer Science

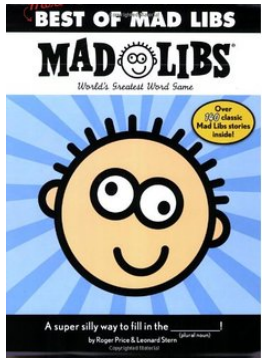
Nov. 29, 2016

Prof. Rodger

Announcements

- Reading and RQ 17 due next time
- Assignment 7 due today
  - Assign 8 due Dec 6
  - APT 9 due Thursday
  - Exam 2 back tomorrow
  - Coming… Points dropped on RQ, Lecture and Lab
- Today:
  - How do you access directories
  - Recursion – Solving problems by solving smaller and smaller similar problems

Lab this week - Madlibs



Rudolph the <adjective>-nosed <noun> had a very <adjective> nose. And if you ever <verb> it, you would really say it glowed.

Rudolph the large-nosed pig had a very crooked nose. And if you ever swim it, you would really say it glowed.

Noun: pig, cow, book, car, hand, lamp, bed, …

Prof Rodger Office Hours today shifted…

- Today Only: 3:30-4:45pm

Math, Engineering, Sociology

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

NETFLIX

Assignment 8: 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 does 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**

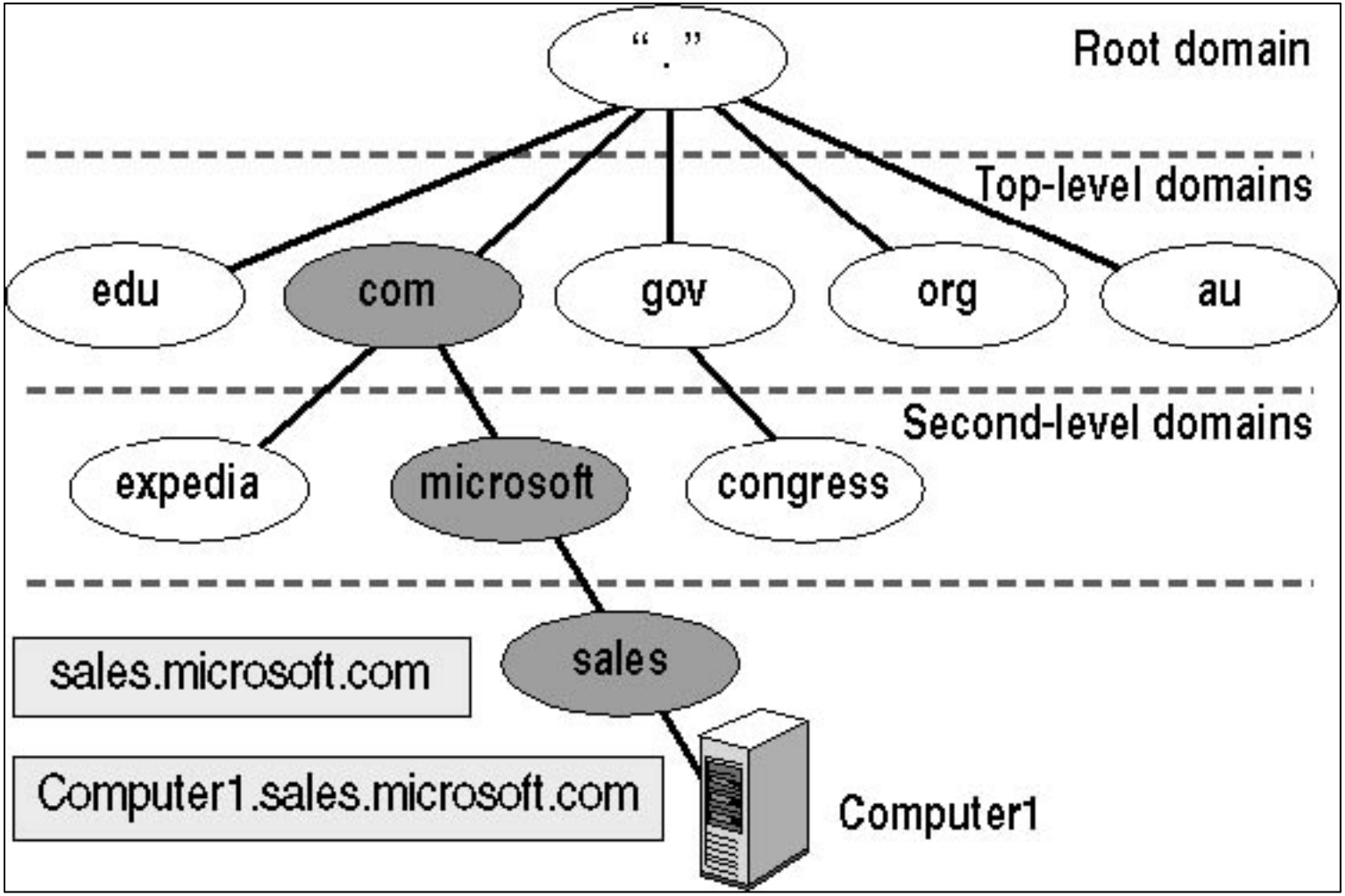
Plan for Today

- Recursion
  - Solving problems by solving similar but smaller problems
- 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 …

Recursion

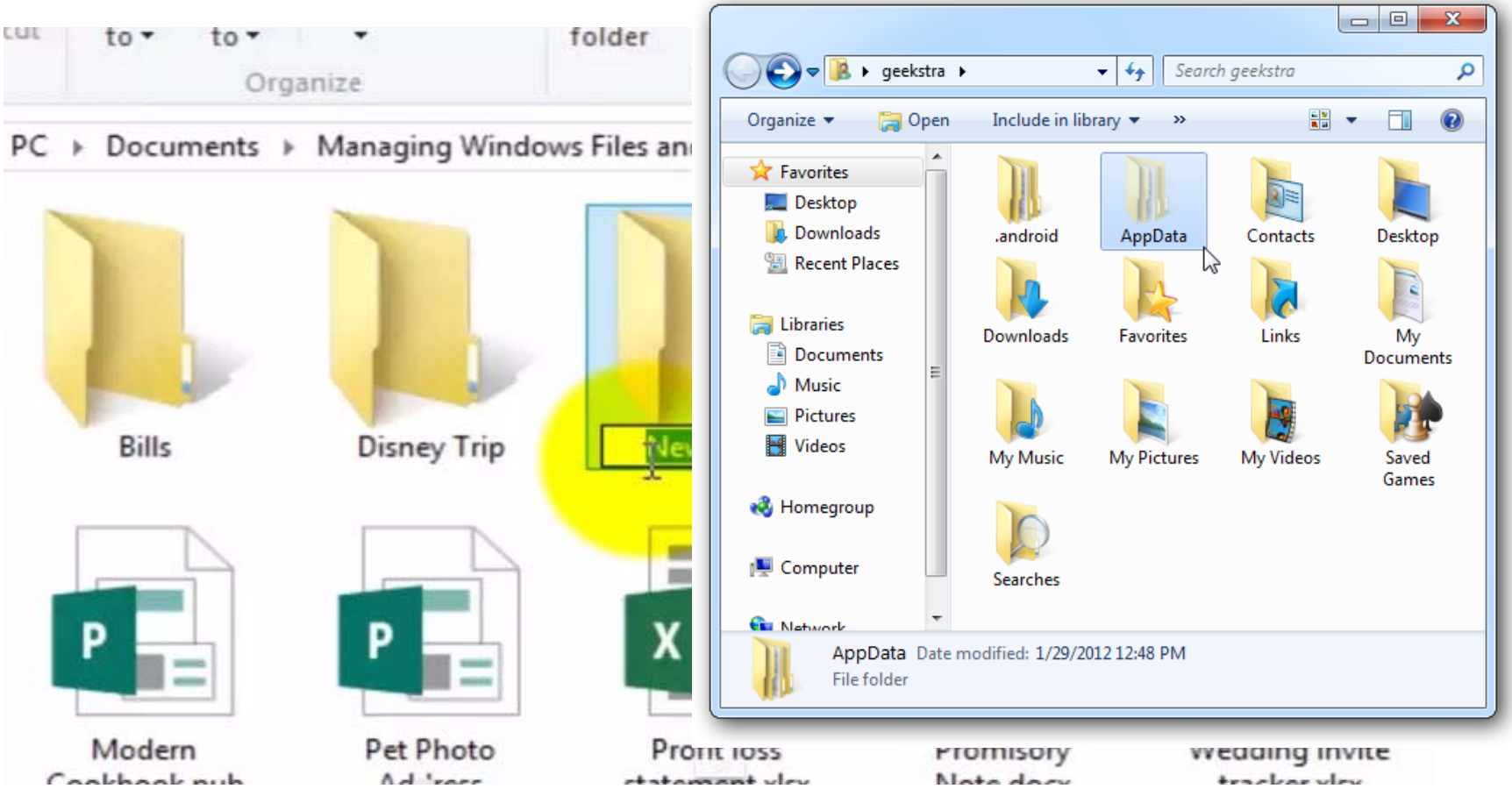
Solving a problem by solving similar but smaller problems

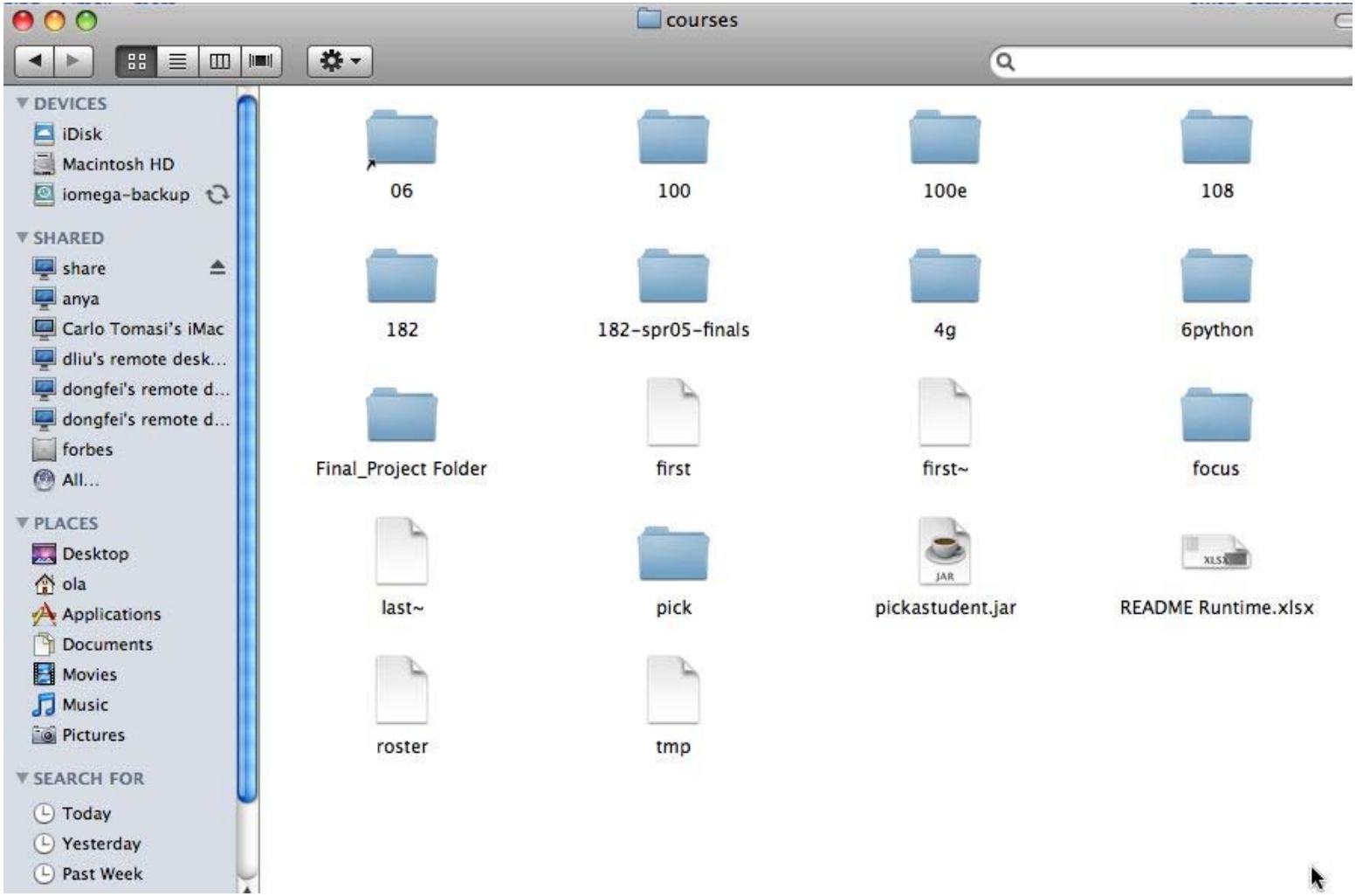
Domain Name System (DNS)



Link: http://computer1.sales.microsoft.com

What's in a file-system Folder?

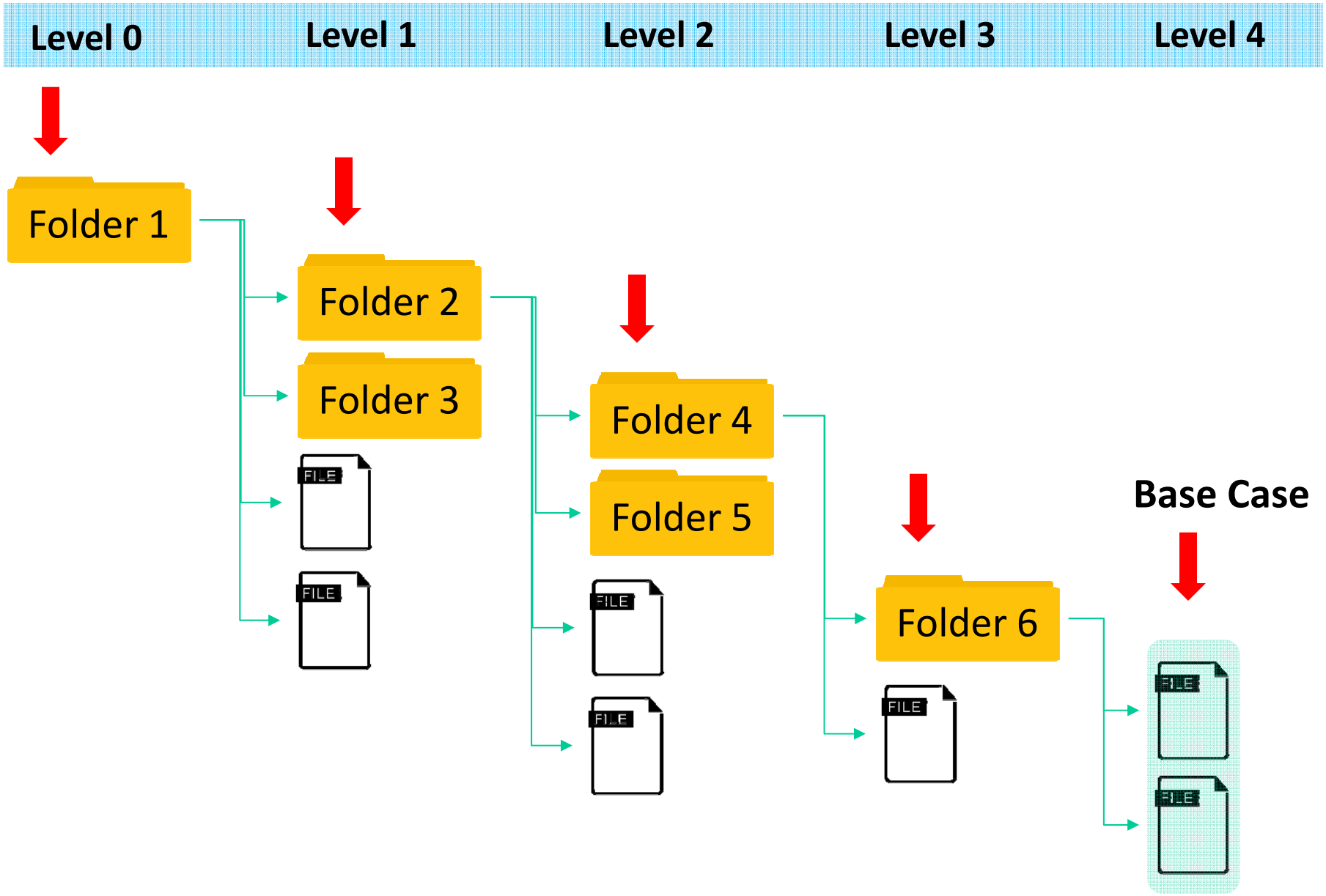



courses
DEVICES
iDisk
Macintosh HD
iomega-backup
SHARED
share
anya
Carlo Tomasi's iMac
dliu's remote desk...
dongfei's remote d...
dongfei's remote d...
forbes
All...
PLACES
Desktop
ola
Applications
Documents
Movies
Music
Pictures
SEARCH FOR
Today
Yesterday
Past Week
06
100
100e
108
182
182-spr05-finals
4g
6python
Final_Project Folder
first
first~
focus
last~
pick
pickastudent.jar
README Runtime.xlsx
roster
tmp

What's in a folder on your computer?

- Where are the **large** files?
- How do you **find them**?
- They take up space!
  - What's the plan –
    1. Erase?
    2. Backup?

Hierarchy in Folder Structure



Recursion to find ALL files in a folder

- A folder can have sub folders and files
- A file cannot have sub files

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

Is that a directory?

If not a directory, it will be a file

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

# on Mac like this:
#bigs = bigfiles("/Users/Susan/Documents",10000)
# on Windows like this:
bigs = bigfiles("C:\\Users\\Susan\\Documents",10000)
```

Example Run

- ('C:\\Users\\Susan\\files\\courses\\cps101\\workspace\\spring2015\\assign4_transform\\data\\romeo.txt', 153088L)
- ('C:\\Users\\Susan\\files\\courses\\cps101\\workspace\\spring2015\\assign4_transform\\data\\twain.txt', 13421L)
- ('C:\\Users\\Susan\\files\\courses\\cps101\\workspace\\spring2015\\assign5_hangman\\src\\lowerwords.txt', 408679L)
- …

Finding Large Files questions
bit.ly/101f16-1129-1

The os and os.path libraries

- Libraries use an API to isolate system dependencies
  - C:\\x\\y **# windows**
  - /Users/Susan/Desktop **# mac**

- 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)`?
  - x = /Users/Susan/Documents
  - y = file1.txt
  - Output = /Users/Susan/Documents/file1.txt

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()`

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

Find large files

If you see a folder,

1. Find the large files and subfolders
2. For the subfolders, repeat the process of finding large files and any other folders within that subfolder
3. Repeat the process until you reach the last folder

Compress or Zip a folder

If you see a folder,

1. Find the files and subfolders
2. For the subfolders, repeat the process of finding files and any other folders within that subfolder
3. At the last stage, start compressing files and move up the folder hierarchy

Structure matches Code

- 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

- Simpler or smaller calls
- Must have a base case when no recursive call can be made
    - Example - The last folder in the folder hierarchy will not have any subfolders. It can only have files. That forms the base case

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, didn't get recursion…..

Inventor of quicksort

Mystery Recursion
bit.ly/101f16-1129-2

```
def Mystery(num):
    if num > 0:
        return 1 + Mystery(num/2)
    else:
        return 2 + num
```

Something Recursion
bitly/101f16-1129-3

```
def Something(data):
    # data is a list of integers
    if len(data) == 0:
        return 0
    if data[0]%2 == 0:  # it is even
        return data[0] + Something(data[1:])
    else:
        return Something(data[1:])
```