10-14 Lecture:

Announcements:

- HW1 released later this week or on monday
- QOTD16 should be easy

QOTD15: solution

```
def findIt(L,x):
    if isinstance(L,list) and len(L) > 0: #censor tail, if instance checks
type
        return(L[0] == x or findIt(L[0],x) or findIt(L[1:],x))
    return False
```

Review word ladder game:

Reading Data From a FIle: Solution 1:

>>>infile = open('words.dat', 'r') read mode
>>>S = infile.read()
>>>len(S)
34542
>>> S[0:20]
'aargh\nabada\nabaci\nah'
>>> W = S.split()
>>> len(W)
5757
W[0:4]
['aargh', 'abaca', 'abaci', 'aback']
>>> infile.close()

Reading Data From a FIle: Solution 2:

```
#alternatively, i could elect to read each line one at a time
>>>infile = open('words.dat', 'r')
>>>s = infile.readline()
```

```
>>>s
>>>'aargh\n'
>>> s = infile.readline().strip() # extra \n at end of word
```

Natural iterative form:

```
infile = open('words.dat', 'r')
W = []
for line in infile:
    W.append(line.strip())
```

Reading data from a file(SS)

Def readWords(filename = ‘words.dat’, ‘r’)

Notice how i use a print statement to give user some feedback when the function is used interactively

```
>>> W = readWords()
5757 words read
```

```
print(str(len(W)) + 'words read'    #bad
```

The format method for strings, .format(), can be used to construct strings by dynamically filling in a template with values
The format operator in essence specifies an entrie language for string output; it has loads of features, although it is poorly described
https://docs.python.org

Basics here

```
print('{} words read.'format(len(W)))
Return W
```

The next step is building word network\
Given a list of words, we now create a second list to represent the network