Announcements

Academic integrity policy description finalized: be sure to read the new version.
http://www.cs.cornell.edu/courses/cs1110/2014sp/about/integrity.php

The aspect students are most likely to not realize:
To not acknowledge the contributions of others is fraudulent. Therefore, to us, the acknowledgment part of your submissions is really important to get right:
"At the top of your assignment, you must list the names of the authors (the students who are submitting the work for course credit) and of every person or other source that contributed to the submission". You don't have to credit course materials or course staff; those resources are understood.
Submission "petitions": email Head TA Dongwook Yoon (DY252), cc:ing profs optional. "Petitions" include requests for extensions due to illness or other reasons, problems with CMS upload, explanations of unusual grouping situations, etc. Petitions are generally considered on a case-by-case business, so send such emails as early as possible.

(We profs need to devote more time to curriculum development and other management issues.)
Example: Palindromes

- String with $\geq 2$ characters is a *palindrome* if:
  - its first and last characters are the same, and
  - the rest of the characters form a palindrome

- All strings with fewer than 2 characters are palindromes

Practical application: RNA secondary structure: loops form because of "antepalindromes" (G/C and A/U)
def ispalindrome(s):
    """Returns: True if string s is a palindrome, False otherwise"""
    # base case

    # recursive case
    A. return s[0] == s[len(s)-1] and ispalindrome(s[1:len(s)-1])
    B. return s[0] == s[len(s)] and ispalindrome(s[1:len(s)])
    C. return s[0] == s[len(s)] and ispalindrome(s[1:])
    D. I did something else
RNA structure and "antepalindromes"

RAGA: RNA sequence alignment by genetic algorithm.
Notredame C¹, O'Brien EA, Higgins DG.
def is5Ahairpin(s):
    """Returns: True if string s represents a hairpin RNA sequence with a loop consisting of 5 As, False otherwise.
    Pre: s is a (possibly empty) string of As, Cs, Gs, and Us."""
    # Assume we have access to function iscomplement(x,y)
A3 motivation:
the Asch social-conformity experiments

Q: Which of A, B, or C has the same length as the line on the card to the left?

College student subject, wearing glasses, leaning forward to look more closely at the cards as many of the other (planted-confederate) students gave incorrect answers.

Distant influence in A3 trial structures

Let \( n_1 \) be a converted node. How might we compute whether another node \( \text{oldone} \) was an influencer, direct or indirect?

**Recursive formulation**: \( \text{oldone} \) is an influencer if it is a direct influencer (i.e., in \( n_1 \)'s contacted_by list), or if \( \text{oldone} \) is an influencer of at least one of \( n_1 \)'s direct influences (again, the nodes in \( n_1 \)'s contacted_by list.)

```python
def was_influenced_by(self, oldone):
    """Returns: True if this node is in oldone's legacy and is actually converted, False otherwise""
```
def was_influenced_by(self, oldone):

    My recursive case most resembles:
    (A) return self.contacted_by[1:].was_influenced_by(oldone)
    (B) for contacter in self.contacted_by:
        return contacter.was_influenced_by(oldone)
    (C) for contacter in self.contacted_by:
        if contacter.was_influenced_by(oldone):
            return True
    Return False
    (D) None of the above