CIS 122

All jumbled up
Playing With Letters
Playing With Letters

● What are anagrams?
  ○ Two words that contain the same letters as each other
  ○ Not necessarily in the same order

● A few notable anagrams
  ○ LISTEN = SILENT
  ○ DORMITORY = DIRTY ROOM
  ○ ELEVEN PLUS TWO = TWELVE PLUS ONE
Playing With Letters

● Constructing anagrams is non-trivial
  ○ [http://words smith.org/anagram/](http://words smith.org/anagram/)
  ○ And constructing sensical anagrams is an art form

● Let's tackle a simpler problem
  ○ How can we tell if two words are anagrams?
  ○ This is still non-trivial

● Just need to see if they contain the same letters
  ○ But how do we do that?
Recursive Step

- Select a letter in one string
  - If it's not in the other string, they're clearly not anagrams
- Remove letter from both words
- See if remaining letters are anagrams
Base Cases

- If strings have different lengths
  - NOT ANAGRAMS

- If a letter in one string isn't in the other
  - NOT ANAGRAMS

- If both strings are empty
  - ANAGRAMS
What pieces do we need?

- Determine if a string contains a character
  - char in string
  - 'a' in 'abcde'

- Remove a character from a string
  - No built-in function
  - We'll need to write our own
String methods

- Methods are special functions called by Python objects
  - `string.method(arguments...)`
  - call method with arguments on string

- `replace(old, new)`
  - Return string with all instances of old replaced with new
    ```python
    >>> "racecar".replace('c', '*')
    'ra*e*ar'
    ```

- `find(char)`
  - Return index of first instance of char in string
    ```python
    >>> "racecar".find('c')
    2
    ```
def remove(string, char):
    """Return new instance of string with first occurrence of char removed"""

    # Find first occurrence of char
    index = string.find(char)

    # Get substrings up to and after char
    upToChar = string[:index]
    afterChar = string[index+1:]

    # Return everything but char
    return upToChar + afterChar
Let's put it all together (to be continued...)

def anagrams(string1, string2):
    """Returns True if strings are anagrams; False otherwise"""

    # Select a letter in one string
    letter = string1[0]

    # If not in other string, not anagrams
    if not (letter in string2):
        return False

    # Remove letter from both words
    # See if remaining letters are anagrams