Using recursion

COSC 101, 2018-04-13

Announcements
- Homework 8 due Thursday

Outline
- Warm-up
- Writing recursive functions

Warm-up
What does every recursive function need?
- Base case
- Recursive case
  - Divide
  - Recurse --- function calls itself
  - Combine

Writing recursive functions
1) Write a recursive function called sum that takes a list of integers and returns the sum of the integers in the list.
   a) What is the condition and result for the base case?
      Condition: list with one (or zero) number(s)
      Result: the number (or zero)
   b) What is the smaller problem on which you recurse?
      List without the first (or last) number
   c) How do you combine the simple piece and result of the recursive call?
      Add the first number in the list to the sum of the rest of the list,
      OR add the last number in the list to the sum of the rest of the list
   d) Write the function.
      ```python
def sum(nums):
    if len(nums) == 0:
      return 0
    else:
      simple = nums[0]
      harder = nums[1:]
      sum_harder = sum(harder)
      return simple + sum_harder
      ```

2) Write a recursive function called reverse that takes a string as a parameter and returns the reverse of the string—e.g., "COLGATE" becomes "ETAGLOC".
   a) What is the condition and result for the base case?
      Condition: string with one character
      Result: single character
   b) What is the smaller problem on which you recurse?
      String without first (or last) character
c) How do you combine the simple piece and result of the recursive call?

Add the reverse of the rest of the string to the first character of the string,
OR add the last character of the string to the reverse of the rest of the string

d) Write the function.

```python
def reverse(string):
    if len(string) <= 1:
        return string
    else:
        simple = string[0]
        harder = string[1:]
        reverse_harder = reverse(harder)
        return reverse_harder + simple
```

3) Write a recursive function called mirror that takes a string as a parameter and returns the string with its mirror--e.g., "COLGATE" becomes "COLGATEETAGLOC".

a) What is the condition and result for the base case?

Condition: string with one character
Result: single character doubled

b) What is the smaller problem on which you recurse?

String without first character

c) How do you combine the simple piece and result of the recursive call?

Concatenate the first character of the string, the mirror of the rest of the string, and the first character of the string

d) Write the function.

```python
def mirror(string):
    if len(string) <= 1:
        return string * 2
    else:
        simple = string[0]
        harder = string[1:]
        mirror_harder = mirror(harder)
        return simple + mirror_harder + simple
```

4) Write a recursive function called is_palindrome that returns True if a string is a palindrome and False otherwise.

a) What is the condition and result for the base case?

Condition: string with one or zero characters
Result: True

b) What is the smaller problem on which you recurse?

String without first and last characters

c) How do you combine the simple piece and result of the recursive call?

Return True if first and last characters are the same and the rest of the string is a palindrome; otherwise False
Write the function.

```python
def is_palindrome(string):
    if len(string) <= 1:
        return True
    else:
        first = string[0]
        last = string[-1]
        middle = string[1:-1]
        middle_palindrome = is_palindrome(middle)
        return first == last and middle_palindrome
```