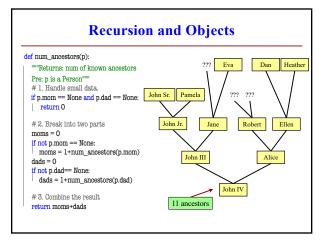


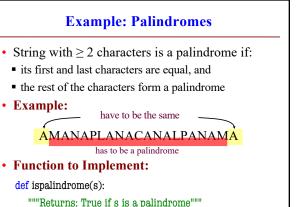
How to Break Up a Recursive Function? def exp(b, c) """Returns: bc Precondition: b a float, $c \ge 0$ an int""" Approach 1 Approach 2 $12^{256} = 12 \times (12^{255})$ $12^{256} = (12^{128}) \times (12^{128})$ Recursive Recursive $b^{c} = b \times (b^{c-1})$ $b^{c} = (b \times b)^{c/2}$ if c even

Raising a Number to an Exponent # of calls def exp(b, c) """Returns: bo 0 0 Precond: b a float, $c \ge 0$ an int""" # b⁰ is 1 4 $\inf c = 0$: 4 return 1 16 32 6 # c > 0if c % 2 == 0: return exp(b*b,c//2) 32768 is 215 b³²⁷⁶⁸ needs only 215 calls! return b*exp(b*b,(c-1)//2)

Recursion and Objects · Class Person (person.py) Dan Heather Objects have 3 attributes name: String mom: Person (or None) dad: Person (or None) Represents the "family tree" Goes as far back as known Attributes mom and dad are None if not known • Constructor: Person(n,m,d) John IV · Or Person(n) if no mom, dad

5 6





8

Example: Palindromes

• String with ≥ 2 characters is a palindrome if:

• its first and last characters are equal, and

• the rest of the characters form a palindrome

def ispalindrome(s):

"""Returns: True if s is a palindrome"

if len(s) < 2:

return True

Halves not the same; not divide and conquer
ends = s[0] == s[·1]
middle = ispalindrome(s[1:·1])

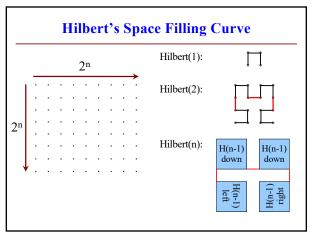
return ends and middle

Recursive Functions and Helpers def ispalindrome2(s): Use helper functions! """Returns: True if s is a palindrome Pull out anything not Case of characters is ignored part of the recursion if len(s) < 2: Keeps your code simple return True and easy to follow # Halves not the same; not divide and conquer ends = $equals_ignore_case(s[0], s[-1])$ middle = ispalindrome(s[1:-1]) return ends and middle def equals_ignore_case(a, b): """Returns: True if a and b are same ignoring case""" return a.upper() == b.upper()

10

9

Example: More Palindromes def ispalindrome3(s): """Returns: True if s is a palindrome Case of characters and non-letters ignored.""" return ispalindrome2(depunct(s)) def depunct(s): """Returns: s with non-letters removed""" if s == ": return s Use helper functions! # Combine left and right Sometimes the helper is if s[0] in string.letters: a recursive function return s[0]+depunct(s[1:]) Allows you break up # Ignore left if it is not a letter problem in smaller parts return depunct(s[1:])



11 12

2