









































Python Programming, 1/e









28





































Example: String Reversal

. def reverse(s):
 if s == "":
 return s
 else:
 return reverse(s[1:]) + s[0]
. >>> reverse("Hello")
 'olleH'

























































Naive Sorting: Selection Sort
 The algorithm has a loop, and each time through the loop the smallest remaining element is selected and moved into its proper position.
 For *n* elements, we find the smallest value and put it in the *Dⁿ* position.
 Then we find the smallest remaining value from position 1 – (*n*-1) and put it into position 1.
 The smallest value from position 2 – (*n*-1) goes in position 2.
 Etc.

Naive Sorting: Selection Sort
 When we place a value into its proper position, we need to be sure we don't accidentally lose the value originally stored in that position.
 If the smallest item is in position 10, moving it into position 0 involves the assignment: nums[0] = nums[10]
 This wipes out the original value in nums[0]!





















95



Python Programming, 1/e

Divide and Conquer: Merge Sort if len(nums) > 1: split nums into two halves mergeSort the first half mergeSort the second half mergeSort the second half merge the two sorted halves back into nums

















































Python Programming, 1/e

121



















