CS115 – Lab 6 – Deconstructing and Constructing Lists – Review

Spring 2018

1  Question 1

Create a function count-even-strings that consumes a (listof Str) and returns a list containing all the strings in the list that have even length.

(count-even-strings (list "ab" "abc" "abcd")) => (list "ab" "abcd").

2  Question 2

Write a function (part-sums L) that consumes a (listof Num), and returns a list of the same length, where each value is replaced by the sum of itself and all the values that come after it.

(part-sums (list 2 3 5 7 9)) => (list 26 24 21 16 9)

Hint
You can tell if a list if empty with the empty? predicate.

(empty? '()) => #true
(empty? (list 2 3 5)) => #false

3  Question 3

One way of sorting a list is to repeatedly swap adjacent out-of-order values in a list, until the list is in order. Each swap increases how sorted it is. After enough swaps, the list will be sorted.

For example, start at the right in (list 11 5 13 7).

• Compare the last two values, 13 and 7. 13 > 7, so swap the last two values, giving (list 11 5 7 13).

• Compare the second from last pair, 5 and 7. 5 < 7, so do nothing; still (list 11 5 7 13).

• Compare the third from last pair, 11 and 5. 11 > 5, so swap these values, giving (list 5 11 7 13).

By “bubbling” through the list once, (list 11 5 13 7) became (list 5 11 7 13), which is better sorted. Bubbling this list again gives (list 5 7 11 13), which is sorted.

Using foldr, write a function bubble that consumes a (listof Num), and returns the result of one pass of swapping out-of-order items in the list, starting at the right.

(bubble (list 1 2 3 4)) => (list 1 2 3 4)
(bubble (list 11 5 13 7)) => (list 5 11 7 13)
(bubble (list 2 6 9 7 4 2 5 7)) => (list 2 2 6 9 7 4 5 7)

Ex.
In the file that contains bubble, type in this program, and test it.
(bsort L) return L, sorted in increasing order.

bsort: (listof Num) -> (listof Num)

Examples:
(check-expect (bsort (list 2 6 9 7 4 2 5 7))
            (list 2 2 4 5 6 7 7 9))

(define (bsort L)
  (foldr (lambda (a b) (bubble b))
         L
         (range 0 (length L) 1)))

Ex. Trace the code, and try to figure out why it works.

4 Question 4

Exercise Write a function (add-back-distance L) that add to each value in L the distance to the end of the list.
For example,
(add-back-distance (list 10 15 20))
=> (list (+ 10 2) (+ 15 1) (+ 20 0))
=> (list 12 16 20)