

Homework Assignment #7

Note

This assignment is due 2:10PM Wednesday, November 21, 2012. Please write or type your answers on A4 (or similar size) paper. Drop your homework by the due time in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

1. (10 points) The following is a *tail-recursive* version of the *length* function (defined in class) that uses an extra component to accumulate the result.

$$\begin{aligned}(\text{len nil res}) &\equiv \text{res} \\(\text{len (cons a y) res}) &\equiv (\text{len y (+ 1 res)})\end{aligned}$$

For any list x , $(\text{length } x) \equiv (\text{len } x \ 0)$. Apply the same technique to define a function that adds all the elements of an integer list.

2. (30 points) Define a Lisp function that sorts a list of numbers, using the partition technique in QuickSort. Pick an element (probably the first) of the list and partition the remainder of the list into two sublists where the first contains all elements smaller than the pivot and the second contains all the other elements. Recursively sort the two sublists and then combine the pivot and the two sorted sublists into a sorted list.
3. (30 points) Sets may be implemented as lists, where all elements are distinct. Define Lisp functions that implement the following operations:
 - (a) Test whether an element is a member of a set.
 - (b) Construct the union of two sets.
 - (c) Construct the intersection of two sets.
4. (30 points) Let `my-reduce`, `my-length`, `my-append`, and `my-remove-if` be Lisp functions as defined in class. Fill in the blanks in the following equivalences.
 - (a) `(my-reduce _____ x _____)` \equiv `(my-length x)`
 - (b) `(my-reduce _____ x _____)` \equiv `(my-append x z)`
 - (c) `(my-reduce _____ x _____)` \equiv `(my-remove-if #'f x)`