CS310: Par. of Prog.
Assgn 1

October 13, 2017

Write the following programs in LISP:

1. Write a recursive function *subst*, that substitutes a given atom x with another y at all levels of a given list, i.e. (subst 'a 'b '(a (a b c) (a d))) returns (b (b b c) (b c)).

2. Write a lisp program to sort a list of lists according to the length of its sublists. We assume that the input is a list that contains elements that are lists themselves. The objective is to sort the elements of this list according to their length. i.e. short lists first, longer lists later, or vice versa.

   Example: (lsort '((A B C) (D E) (F G H) (D E) (I J K L) (M N) (O)))
   ((O) (D E) (D E) (M N) (A B C) (F G H) (I J K L))

3. Insertion sort

4. Write a function to transpose a matrix represented as list of lists, i.e. (trans '((a b) (c d)) should return ((a c) (b c)).

5. Write a function that takes a function as input and outputs a memoized version of it, i.e. if the function is called twice on the same input, then it doesn't evaluate the function again but outputs the answer from its memory.

6. Please write an efficient version of a function to compute the n-th fibonacci number. ($O(n)$).
   Hint: Use tail-recursion.