CSCI141 Nov 25 2019

Problem 1: Write pseudocode for the following function. Do not use any `list` methods.

```
def find(v, lst):
    """ Return the index of the first
        occurrence of v in lst.
        Return -1 if v is not in the list.
        Precondition: lst is a list. """
```

Problem 2: Write pseudocode for the following function without using any `list` methods. Try to do it as *efficiently* as possible: compare v to as few elements of `lst` as possible. Can you find v (or determine it's not in the list) using fewer than `len(lst)` comparisons?

```
def find(v, lst):
    """ Return the index of the first occurrence of v in lst.
    Return -1 if v is not in the list.
    Precondition: lst is a list of things that can be compared
    with the < operator, and is in sorted order
    (i.e. lst[i] <= lst[i+1] for all i in range(len(lst)-1) """
```

Problem 3: Write pseudocode for the following function without using any `list` methods.

```
def sort(lst):
    """ Sort the given list.
    Precondition: lst is a list of things that can be compared
         with the < operator.
    Postcondition: lst[i] <= lst[i+1] for all i in
         range(len(list)-1), or in other words, lst is sorted
    """
```