CSCI141 Nov 20 2019

To the right of each program, draw the memory diagram for the program as it is executed.

```
def z1(a_list):
    a_list[0] = 0

a = [1, 1, 1]
z1(a)
print(a)
```

```
def z2(a_list):
    a_list = []

a = [1, 1, 1]
z2(a)
print(a)
```

```
def z3(x):
    a_list = [x, x, x]
    return a_list

b = 2
a = z3(b)
print(a)
```

Coding Problem 1: Implement 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. """
```

Coding Problem 2: Implement 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) """
```