Quiz #10: Common Programming Mistakes

Since you have just gotten a lot of practice with Big-Oh questions on pencil-and-paper, this quiz will focus on a different topic: identifying and fixing common programming errors we’ve encountered in this course.

1. Here is a function which is supposed to delete the first item from a list (it’s a mutating function), and a sample use of it.

```python
def remove_first(lst):
    lst = lst[1:]

> lst = [1, 2, 3]
> remove_first(lst)
> lst
```

Explain what the output on the last line is, and why it occurs. Then, fix the function implementation so that it works as expected.

2. Here is a stack function which computes the sum of a stack of numbers, but should not mutate the stack, and a sample use of it.

```python
def sum_stack(stack):
    s = 0
    copy = stack
    while not copy.is_empty() :
        s += copy.pop()
    return s
```

Explain the problem with this function, and then fix the implementation so that it works correctly.
3. Here is a LinkedList method which returns whether the list contains an item bigger than 10, and a sample usage.

```python
class LinkedList:
    ...

    def search_big(self):
        if self._first is None:
            return False
        else:
            curr = self._first
            while curr is not None:
                if curr.item > 10:
                    return True
                else:
                    return False
                curr = curr.next

> lst1 = LinkedList([20, 10, 30])
> lst1.search_big()
True
> lst2 = LinkedList([10, 20, 30])
> lst2.search_big()
False
```

Explain the above output, identifying the problem with the above code. Then, write a correct implementation.
4. Here is some code (body of class is omitted, but assume it works correctly).

class A:
    ...

def my_f(a, y):
    """@type a: A"
    x = a.big - a.get_small(y)
    return x + 10

Sadly, when David runs the following code, he gets an error:

> a = A(10, 20, 30)
> a.my_f(17)
AttributeError: 'A' object has no attribute 'my_f'

But it’s right there in the code! Explain the problem, and how to fix it.

5. Here is a mutating Tree method which adds a number to the root item of the tree, and sample usage.

class Tree:
    ...

    def add_num(self, num):
        if not self.is_empty():
            self._root = self._root + num

> t = Tree(5)
> t = t.add_num(10)
> t._root

Sadly, an error occurs. Explain the problem, and how you would fix it.