Name: _____

You have 20 minutes.

Problem 2.1   Points: 2+2+3+3

Consider the following algorithm:

```
fun foo(l : List[int], m : List[int]) : List[Int] =
    x := l
    y := m
    while y ≠ []
        x := append(x, y.head)
        y := y.tail
    return x
```

List is implemented as an immutable linked list, and *append* adds an element at the end of a list.

1. Which operation does foo implement?

2. Assuming that both inputs have the same length n, give the Θ-class of the time complexity of foo in terms of n?

3. Give a loop invariant $F(l, m, x, y)$ for the while-loop with which we can prove foo partially correct.

4. Argue informally why the while-loop terminates (2 points), **OR** give a formal termination ordering $T(l, m, x, y)$ for the while-loop (3 points).

Problem 2.2   Points: 2+2+3+3

Consider the following algorithm operating on a mutable list x:

```
fun foo(x : List[int]) =
    low := 0
    high := length(x) − 1
    while low < high
        l := get(x, low)
        h := get(x, high)
        update(x, low, h)
        update(x, high, l)
        low := low + 1
        high := high − 1
```

1. Which operation does foo implement?

2. Assuming that *List* is implemented as an array, give the Θ-class of the time complexity of foo (in terms of the length n of x).

3. Assuming that *List* is implemented as a linked list, give the Θ-class of the time complexity of foo (in terms of the length n of x).

4. Argue informally why the while-loop terminates (2 points), **OR** give a formal termination ordering $T(x, low, high)$ for the while-loop (3 points).