CSc 120, Spring 2019
Discussion Section 8

SOLUTIONS

Problem 1

a) O(n), n is an input integer
b) $O(n^2)$, n is the argument of fun2
c) O(n), n is the length of numlist
d) Various solutions

Problem 2

a) Version 1

```
def has_dups(alist):
    for i in range(len(alist)):
        for j in range(len(alist)):
            if i != j and alist[i] == alist[j]:
                return True
    return False
```

The complexity of the function is $O(n^2)$.

b) Version 2

```
def has_dups(alist):
    counts = {}
    for elem in alist:
        if elem in counts:
            return True
        else:
            counts[elem] = 1
    return False
```

c) To determine the complexity of the solution above, we would need to know the complexity of the operation

```
elem in counts
```

which is *looking up an element in a dictionary*, and the complexity of

```
counts[elem] = 1
```

which is *inserting an element into a dictionary*. Note: Later in the semester, we will explore the implementation of dictionaries (which are called hash tables in most other languages) and discover that both of these operations are O(1).

Problem 3

Two different exceptions are caught by the except block. A divide by zero error would cause the error message to be printed that relates to opening a file. This is confusing and ambiguous.

Problem 4

List.insert() : O(n), because an insertion requires shifting O(n) elements
List.append(): O(1), because the address of the next appended element is known