Weeks 1 - 7 Review

Week 2

Reasoning Over Code

```
def f(x, y, z):
    assert(x + y + z < 20)
    prev = None
    total = 10**x  # note: total is not starting at 0
    for w in range(x, y, z):
        if (prev != None): assert(w - prev == x)
        prev = w
        total += w
    return (total == 1018) and (y % x == 1)
```

nthSquareNumber(n) - Write the function nthSquareNumber that takes in a positive integer n and returns the nthSqaureNumber >= 1. A square number is defined by having an integer square root.

Week 3

```
def ct1(s, t):
    for c in s:
        if (c.upper() not in "NO!!!"):
            i = t.find(c)
            print(i, s[ i ], t[ i ])
ct1("net", "woot")
```

Free Response: Word Wrap

Write the function wordWrap(text, width) that takes a string of lowercase text and a positive integer width, and returns a possibly-multiline string that matches the original string, only with line wrapping at given width. So wordWrap("abc", 3) just returns "abc", but wordWrap("abc", 2) returns a 2-line string, with "ab" on the first-line and "c" on the second line. After you complete word wrapping in this way: all spaces at the start

and end of each resulting line should be removed, and then all remaining spaces converted to dashes.

```
assert(wordWrap("abcdefghij", 4)   ==   """\
abcd
efgh
ij""")
assert(wordWrap("a  b    c    de   fg", 4)   ==   """\
a-b
c-de
fg""")
```

Week 4

-write each sort from scratch (iteratively)

-Selection Sort

-Bubble Sort

-Merge Sort

- case studies: locker problem, sieve.

Week 5

Code Tracing:

```
import copy

a = [ [15, 112], ["is", 'fun'] ]
b = a
c = copy.copy(a)
d = copy.deepcopy(a)

print(a is b, a is c, a is d, c is d)
print(a == b, a == c, a == d, c == d)

a[0] = ["nothing", "else"]
b[1] = [42, "foo"]
c[0] += [15, 122]
d[1] += ["Jordan", ":)"]

c += ["koz", "mjs"]
```

```
print(a, b, c, d)
```

-High-level word search

Week 6

-Identifying Big-Oh (w/ sorts)
-high level hashing
-sets/dictionary FR

Questions:

Find the Big-Oh:

def f1(L):

```
# Assume L is a non-empty list of integer values
     a = copy.copy(L)
     n = len(a)
     for  in range(n):
          minIndex = startIndex
          for i in range(startIndex, n):
               if (a[ i ] < a[minIndex]):
                              minIndex = i
          swap(a, startIndex, minIndex) # Big-Oh → O(1)
     for i in range(len(a)):
          if (a[ i ] != i):   return    False
     return    True
```

What is the Big-Oh of Each Sort? Why (Proof)?

Big-Oh of Sets and Dictionaries:
-Adding an element to a set

Free Response: rowSetMap(a)
Write the function rowSetMap(a) that takes a non-empty rectangular 2D list a(only having integers), and returns a

dictionary mapping each value in the 2D list to a set of indexes of rows in which they appear.

```
 a = [[2, 4, 6],
        [5, 5, 6]]
assert(rowSetMap(a) == { 2:set([0]), 4:set([0]),
                                 5:set([1]), 6:set([0, 1]) })
```