Name:_________________________________ Section:___ Andrew Id: ____________________ a

15-112 Fall 2019 Quiz 8a

*** Up to 20 minutes. No calculators, no notes, no books, no computers. * Show your work!**

*** No recursion**

1. Short answers: [20 pts]

a) Given the list [5, 2, 7, 3], what will the list be after exactly 2 swaps are made in selectionSort, as it works in xSortLab (which selects the maximum value on each pass)?

b) State and explain the Big-O of selectionSort with no more than two short sentences **or** a well-labeled diagram. (Use only the space below. If you need more, you've written too much.)

c) You notice that **often** isPrime(n) and fasterIsPrime(n) take approximately the same amount of time to run. Briefly, does this mean they both have the same Big-O? Why or why not?

d) Copies and swaps both assign values to list indices. Does mergeSort typically perform fewer, greater, or an equal number of assignments compared to selectionSort?

e) Put the following function families in order, from the slowest runtime to the fastest:

O(logn), O(2**n), O(n), O(n**0.5), O(n**3)

Slowest **Fastest**

2. Big-O runtimes: [10 pts]
State the overall Big-O of each of these two functions in the boxes to the right

```
def bigOh1(L): #L is a list
    n = len(L)
    for i in range(n//8, n//4, n//2):
        for j in range(0, n**2, 2):
            L.append(i)
    return L
```

```
def bigOh2(L): #L is a list
    while len(L)>1:
        L = L[:len(L)//2]
    return L
```

3. Free Response: hasSquarePair(L) [30 pts]
Write the function hasSquarePair(L) that takes a list of positive integers and returns True if the list contains an integer whose square is also in the list. So [4,3,6,12,9] returns True because 3**2 is 9, which is also present in the list. Likewise, the list [4,7,5,22,30] returns False because the list contains no integer whose squared value is also in the list. **You must annotate each line with its Big-O runtime, and your function must have an efficiency of O(N) to receive full credit!**

4. Free Response: popularityScores(d) [30 pts]

Free Response: Recall that friendsOfFriends(d) takes a dictionary d like this, which indicates the friends of Fred and the friends of Wilma and the friends of Barney:

```
d = dict()
d["fred"] = set(["wilma", "betty", "barney"])
d["wilma"] = set(["fred", "betty", "dino"])
d["barney"] = set(["fred", "betty"])
```

Write the function popularityScores(d) that takes a dictionary of that form and returns a new dictionary which indicates for each person how many people consider that person a friend. For example, given d above:

```
popularityScores(d) == {"wilma" : 1,
                        "betty" : 3,
                        "barney" : 1,
                        "fred" : 2,
                        "dino" : 1}
```

Note: You do not need to include anyone who is no one's friend. They forgive you.

Don’t forget the CT on the last page!

5. Code Tracing [10 pts]
Indicate what this prints in the box to the right:

```
def ct1(M,x):
    L = 0
    R = len(M) - 1
    while L <= R:
        i = (L + R) // 2
        if M[i] == x:
            return i
        elif M[i] < x:
            L = i + 1
        else:
            R = i - 1
        print(M[L:R+1])
    return -1

lst = [8,10,11,12,14,17,18]
print(ct1(lst,14))
```

6. Bonus/Optional: Code Tracing [2 pts]
Indicate what this prints. Very clearly circle your answer (and nothing else):

```
def bonusCT1(s = 'Maine has 14 counties, and 432 towns.'):
    def f(s):
        s, d, prevc = s.replace(' ',''), dict(), 'x'
        for c in s: d[c], prevc = prevc, c
        return ''.join([d.get(str(n), '') for n in range(10)])
    while (len(f(s)) > 1): s = f(s)
    return s
print(bonusCT1())
```