Quiz 7 MC, CT, ROC (40 pts)

Time Limit: **12 mins**

Name: _____ AndrewID: _____

This part does not contain any free responses. Do not run any code when completing this part of Quiz 7.
Time limit: 12 mins

Part 1 — Multiple Choice (20 pts total)

1. (1 pts) How many passes does mergesort do on a list of length N?

- ◯ `O(N)`
- ◯ `O(N**2)`
- ◯ `O(logN)`
- ◯ `O(NlogN)`

2. (1 pts) How many steps per pass does mergesort do on a list of length N?

- ◯ `O(N)`
- ◯ `O(N**2)`
- ◯ `O(logN)`
- ◯ `O(NlogN)`

3. (2 pts) If we need to do 10 searches in a list `L`, is it faster to linear search 10 times, or sort once and then binary search 10 times?

- ◯ Linear search 10 times
- ◯ Sort once and then binary search 10 times
- ◯ They take the same amount of time

4. (2 pts) If we need to do N searches in a list `L`, is it faster to linear search N times, or sort once and then binary search N times?

- ◯ Linear search N times
- ◯ Sort once and then binary search N times
- ◯ They take the same amount of time

5. (2 pts) If on a given computer a function takes 30 seconds to run on a list with 1000 values and 120 seconds to run on a list with 2000 values, what would you expect the efficiency of the function to be? Assume N is the length of the list.

- ◯ `O(N)`
- ◯ `O(logN)`
- ◯ `O(N**2)`
- ◯ `O(N**3)`

6. (4 pts) What is the worst case efficiency of the following function? Assume `L` is a list of length N.

```
def f(L):
   A = []
   for _ in L:
       i = len(L) - 1
       while i > 0:
           A.append(L[i])
           i //= 2
   return A
```

- ◯ `O(N)`
- ◯ `O(logN)`
- ◯ `O(NlogN)`
- ◯ `O(N**2)`

7. (4 pts) What is the worst case efficiency of the following function? Assume `s` is a string of length N.

```
def f(s):
   a = set()
   b = set()
   for v in s:
       if v in a:
           b.add(v)
       else:
           b = set(s)
       a.add(v)
   return b
```

- ◯ O(N)
- ◯ O(NlogN)
- ◯ O(N**2)
- ◯ O(N**3)

8. (4 pts) What is the efficiency of the following function? Assume `L` is an NxN 2D list.

```
def f(L):
    N = len(L)
    s = 0
    for rowList in L:
        a = sorted(rowList)
        s += a[N//2]
    return s
```

- ◯ O(N)
- ◯ O(NlogN)
- ◯ O(N**2)
- ◯ O(N**2 * logN)

Part 2 — Code Trace / Reasoning Over Code (20 pts total)

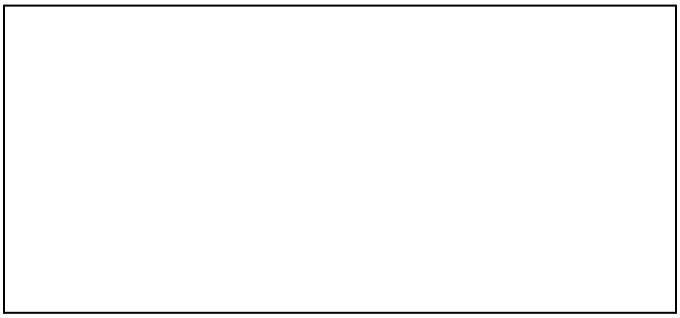
Code Trace 1 (10 pts)

What does the following code print?

```
def ct(L):
    s = set()
    t = set()
    for x in L:
        print(s | x)
        t = x - s
        s |= x
        if (s & t) == s:
            print('wow')
    return s.intersection(t)

print(ct([{1, 2}, {2, 3, 4}]))
```

Output:



Reasoning Over Code 1 (10 pts)

Fill in the blanks in function g so that f(d) == g(d) for all valid inputs d.

```
'''
def f(d):
    i = 0
    j = None
    k = None
    for key in d:
        i += 1
        val = d[key]
        if key == val:
            j = val if j == None else min(val, j)
            k = val if k == None else max(val, k)
    return i, j, k

def g(d):
    L = []
    for key, val in d.items():
        if key == val:
            L.append(_____________________) # blank1
    i = _____________________ # blank2
    j = None if L == [] else _____________________ # blank3
    k = None if L == [] else _____________________ # blank4
    return i, j, k
'''
```

This page intentionally left blank for scratch work

This page intentionally left blank for scratch work