

1. Consider the following function.

```
def q1(L):
    n = len(L)
    count = 0
    for i in L:
        if i == max(L) or i < 0:
            count = count + 1
    return count
```

- a. What is the Big-O running time bound for q1 in terms of n, where $n == \text{len}(L)$?
- b. Is q1's growth rate exponential? Why or why not?
- c. The algorithm can be made faster (while still computing the same thing) via small changes in some or all of the places marked ??? Write updated code next to the ???'s (you may not change any other parts of the code).

```
def q1faster(L):
    n = len(L)
    count = 0
    ???
    for i in L:
        ???
        ???
    return count
```

- d. What is the Big-O running time bound for q1faster?

2. What is the value of each of the following expressions?

a. `len([c for c in 'computer science' if c not in 'aeiou'])`

b. `[('$'+ str(i) + '.99') for i in range(1,5)]`