

Problem Set 0

Due January 18

1. [19] Consider this pseudocode for sorting a list of 0s and 1s so that all 0s are before all 1s.

```
n ← length of L
while true do
    for i from 1 to n do
        if L[i] = 1 then
            break
    for j from n to 1 do
        if L[j] = 0 then
            break
        if j > i then
            swap L[i], L[j]
        else
            break
```

- [2] Give a high-level description of the algorithm. It should be as short and clear as possible. You should avoid details that matter for runtime but not for correctness.
- [17] Analyze the runtime of the pseudocode, when the input is a list of length $2n$ containing n 0s and n 1s. Fill out the table with O , Ω , and Θ bounds on the best-, worst-, and average-case runtime. The distribution for the average-case runtime is the uniform distribution on lists with n 0s and n 1s.

Give appropriate justifications. You don't need a formal proof for every entry. Some entries may not need any statement at all. The easy entries are worth 1 point, and the ones requiring more complex arguments are worth 3.

	O	Ω	Θ
best-case			
worst-case			
average-case			

- [1 (Bonus)] Give an implementation of the same algorithm with better worst-case and average-case runtime.

2. [10] The following Python code gets a string containing a number, and returns a string with a list of all numbers up to the input number, separated by commas. What is the runtime?

For this question, you are allowed (and encouraged!) to look up information online. Be careful, this one is tricky.

```
def list_of_numbers(num):
    n = int(num)
    out = ""
    for i in range(n):
        out += str(i+1) + ","
    return out
```