DSC 40B - Homework 01

Due: Wednesday, April 10

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 p.m.

Problem 1.

Suppose

- algorithm A takes n milliseconds to run on a problem of size n,
- algorithm B takes n^2 milliseconds to run on a problem of size n,
- algorithm C takes 2^n milliseconds to run on a problem of size n,
- algorithm D takes $\log_2(n)$ milliseconds to run on a problem of size n.

What is the largest problem size each algorithm can solve in 1 second, 10 seconds, and 1 minute? That is, fill in the following table:

	1 sec.	10 sec.	1 min
A	?	?	?
B	?	?	?
C	?	?	?
D	?	?	?

Problem 2.

Determine the asymptotic time complexity of the following piece of code, showing your reasoning and your work.

```
def f(n):
    i = 1
    while i <= n:
        i *= 2
        for j in range(i**2): # <-- note the range!
            print(i, j)
```

Hint: you might need to think back to calculus to remember the formula for the sum of a geometric progression... or you can check wikipedia.[1]

Problem 3.

Consider the code below:

```
def foo(n):
    i = 1
    while i**3 < n - 100:
        i += 1
    return i
```

a) What does `foo(n)` compute, roughly speaking?

[1] `https://en.wikipedia.org/wiki/Geometric_series`

b) What is the asymptotic time complexity of `foo`?