Section 2 (Algorithm Analysis)

8. (6 points) For each of the following statements, circle T if it is true and F if it is false.

   - $n^2 + n \log n$ is $\Theta(n^2)$: T or F
   - 2^n is $O(n!)$: T or F
   - $n \log n$ is $\Omega(n^2)$: T or F
   - $(\log n) \cdot (\log n)$ is $O(n^2)$: T or F
   - There is an algorithm with worst-case runtime that is $O(n^2)$ and best-case runtime that is $\Omega(n^3)$: T or F
   - There is an algorithm with worst-case runtime that is $O(n^2)$ and best-case runtime that is $O(n^3)$: T or F

9. (4 points) Suppose you have an algorithm with the six running times listed below. (Assume these are the exact number of operations performed as a function of the input size n, not asymptotic running times.) Suppose you have a computer that can perform 10^{10} operations per second, and you need to compute a result in at most an hour of computation. For each of the algorithms, what is the largest input size n for which you would be able to get the result within an hour?

   (a) $100n^2$

   (b) $\log n$

   (c) 2^n

10. (3 points) In words, what is the definition of the worst-case runtime for an algorithm?

11. (3 points) Give a function $f(n)$ such that the worst-case runtime of the following algorithm is $\Theta(f(n))$. Recall that $\lfloor x \rfloor$ takes the *floor* of x, meaning that it rounds down to the nearest integer.

```
Algorithm-1(array A of length n):
  Result = 0
  i = n
  While i > 10:
      For j in 1 to n:
          Add A[j] to Result
      Set i = ⌊i/2⌋
  Return Result
```

12. (4 points) Give a function $f(n)$ such that the worst-case runtime of the following algorithm is $\Theta(f(n))$.

```
Algorithm-2(array A of length n):
  Result = 0
  For i in 1 to n:
      For j in 2^i to n:
          Add A[j] to Result
  Return Result
```