This problem set is individual and worth a total of 100 points. Solutions **must** be handed in using your account on Gradescope in PDF form by Sunday 2/9 at 11:59pm. Please write your answers **clearly within the space provided** for each question.

1. Simplify the following:

   (a) [3 points] $\log_2 xy^2 - \log_2 x^2 - 2\log_2 y$

   (b) [3 points] $\log_2 16x^2$

   (c) [3 points] $\log_3(9x^4) - \log_3(3x)^2$

2. [5 points] Rewrite the following expression into its closed form (i.e. without the sigma): $\sum_{i=1}^{n}(2+i)$.

3. [28 points] For each of the following, give a formula $T(n)$ for the exact number of instructions:

   (a)
```
for (int i = 0 ; i < n ; i++) {
    for (int j = 0 ; j < n ; j++) {
        for (int k = 0 ; k < n ; k++) {
            // count 1 instruction
        }
    }
}
```

   (b)
```
for (int i = 0 ; i < n ; i += 4) {
    // count 1 instruction
}
```

(c)

```
for (int i = 0 ; i < n ; i++) {
    // count 1 instruction
    for (int j = 0 ; j < n ; j++) {
        // count 1 instruction
    }
}
```

(d)

```
// pow C++ function pow(base, exponent)
for (int i = 1 ; i < pow(2, n) ; i *= 2) {
    // count 1 instruction
}
```

(e)

```
for (int i = n ; i > 1 ; i /= 2) {
    // count 1 instruction
}
```

(f)

```
for (int i = 0 ; i < 4n ; i++) {
    for (int j = 0 ; j < i ; j++) {
        // count 1 instruction
    }
}
```

(g)

```
for (int i = 0 ; i < n*n ; i++) {
    for (int j = 0 ; j < i ; j++) {
        // count 1 instruction
    }
}
```

4. [15 points] Rank the following functions by their asymptotic growth rate in ascending order (all log functions are base 2):

$T(n)$	Rank	$T(n)$	Rank	$T(n)$	Rank
$50n \log n$		2^{100}		$\log \log n$	
$\log^2 n$		$2^{\log n}$		2^{2^n}	
$\lceil \sqrt{n} \rceil$		$n^{0.01}$		$1/n$	
$4n^{3/2}$		4^n		$n^2 \log n$	
$4^{\log n}$		$\sqrt{\log n}$		$\sqrt{81}$	

5. [15 points] Mark each of the following as true or false.

$T(n)$	Big O	T/F	Big Omega	T/F	Big Theta	T/F
$n^2/10 + 10n \log n$	$O(n \log n)$		$\Omega(n \log n)$		$\Theta(n \log n)$	
$2n^2 + n \log n$	$O(n^2)$		$\Omega(n)$		$\Theta(\log n)$	
$(n/2) logn + 4n$	$O(2^n)$		$\Omega(n \log n)$		$\Theta(n \log n)$	
$10\sqrt{n} + 2 \log n$	$O(\log n)$		$\Omega(n)$		$\Theta(\log n)$	
$3\sqrt{n} + 10 \log n$	$O(\sqrt{n})$		$\Omega(1)$		$\Theta(\sqrt{n})$	

6. [15 points] Complete the following table.

$T(n)$	Big Theta
$\log n + 200n \log n$	
$2^n + n^2$	
$\sqrt{n} + \log n$	
$2n + 3n + 4n + 5n + 6n$	
$\sqrt{n} + 10 \log n$	

7. [6 points] An array `A` contains $n-1$ unique integers in the range $[0, n-1]$; that is, there is one number from this range that is not in `A`. Describe an $O(n)$ time algorithm for finding that number. You are allowed to use only $O(1)$ additional memory besides the array `A` itself. Provide a clear and precise explanation of the algorithm.

8. [7 points] Suppose that each row of an $n \times n$ matrix `M` consists of only binary digits, such that, in any row of `M`, all the 1's come before any 0's in that row. Assuming `M` is already in memory, describe an $O(n)$-time algorithm for finding the row of `M` that contains the most 1's. Provide a clear and precise explanation of the algorithm.