1. Using the rules given in the course notes, perform an exact count complexity analysis of the body of the following function. (Your answer should be a function of the parameter N.)

```
void Mystery(int M[N][N], const int N) {
    for (int R = 0; R < N; R++) {  // 1 before loop, 2 per pass, 1 at end
        for (int C = 1; C < N; C++) {  // 1 before loop, 2 per pass, 1 at end
            if (M[R][C-1] < M[R][C])  // 4 index ops + 1 arithmetic op +
                // 1 comparison per pass
                M[R][C-1] = M[R][C];  // 4 index ops + 1 arithmetic op
                // + 1 assign
            else {  // 4 index ops + 1 arithmetic op
                    // + 1 assign
                M[R][C] = M[R][C-1];  // 4 index ops + 1 arithmetic op
                // + 1 assign
                M[R][C-1] = 0;  // 2 index ops + 1 arithmetic op
                // + 1 assign
            }
        }
    }
}
```

Note: array index operations count 1 time unit each.

Applying the rules, we get the complexity formula:

\[
T(N) = 1 + \sum_{R=0}^{N-1} \left( 1 + 1 + \sum_{C=1}^{N-1} \left( 1 + 6 + \max(6,10) + 1 \right) + 1 + 1 \right) + 1
\]

Simplifying:

\[
T(N) = 1 + \sum_{R=0}^{N-1} \left( 1 + 1 + \sum_{C=1}^{N-1} \left( 1 + 6 + \max(6,10) + 1 \right) + 1 + 1 \right) + 1
\]

\[
= 2 + \sum_{R=0}^{N-1} \left( 4 + \sum_{C=1}^{N-1} 18 \right)
\]

\[
= 2 + \sum_{R=0}^{N-1} \left( 4 + 18(N-1) \right) = 2 + \sum_{R=0}^{N-1} \left( 18N - 14 \right)
\]

\[
= 2 + \sum_{R=1}^{N} \left( 18N - 14 \right) = 2 + N(18N - 14)
\]

\[
= 18N^2 - 14N + 2
\]
2. For each of the following inequalities, what is the smallest value of $N$ such that the inequality holds for all $n \geq N$?

(a) $0.1n \geq 10 \log n$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$0.1n$</th>
<th>$10 \log n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>980</td>
<td>98.0</td>
<td>99.36638</td>
</tr>
<tr>
<td>990</td>
<td>99.0</td>
<td>99.51285</td>
</tr>
<tr>
<td>995</td>
<td>99.5</td>
<td>99.58553</td>
</tr>
<tr>
<td>996</td>
<td>99.6</td>
<td>99.60002</td>
</tr>
<tr>
<td>997</td>
<td>99.7</td>
<td>99.61450</td>
</tr>
<tr>
<td>998</td>
<td>99.8</td>
<td>99.62896</td>
</tr>
<tr>
<td>1000</td>
<td>100.0</td>
<td>99.65784</td>
</tr>
</tbody>
</table>

Plotting values for each function, we get the table at right. So, $N = 997$

(b) $\frac{1}{2}n^2 - n \leq 20n \log n$

Trick question… the LHS is $\Theta(n^2)$ and the RHS is $\Theta(n \log n)$, and $n^2$ is strictly $O(n \log n)$.

So, there isn’t any point after which the LHS is always less than the RHS.

3. Divide the following functions into non-overlapping categories ($\Theta$ equivalence classes), so that two functions, say $f(n)$ and $g(n)$, are in the same category if and only if $f(n)$ is $\Theta(g(n))$. Arrange the categories from the lowest order of magnitude to the highest. A function may be in a category by itself, or there may be several functions in the same category.

<table>
<thead>
<tr>
<th>$f(n)$</th>
<th>$\log n$</th>
<th>$n + \log n$</th>
<th>$n^2$</th>
<th>$n^{0.3}$</th>
<th>$n^{0.3} \log n$</th>
<th>$n^3$</th>
<th>$n^3 \log n$</th>
<th>$2^n$</th>
<th>$(n^2 + 4)^{1/2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>$\log n^5$</td>
<td>$\log n^2$</td>
<td>$n + \log n$</td>
<td>$n^3$</td>
<td>$n^{0.3}$</td>
<td>$n^{0.3} \log n$</td>
<td>$n^3$</td>
<td>$n^3 \log n$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>$\log n$</td>
<td>$n^5$</td>
<td>$\log n^2$</td>
<td>$n + \log n$</td>
<td>$n^3$</td>
<td>$n^{0.3}$</td>
<td>$n^{0.3} \log n$</td>
<td>$n^3$</td>
<td>$n^3 \log n$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>$n^2 \log n$</td>
<td>$n^{0.3}$</td>
<td>$n^{0.3} \log n$</td>
<td>$n^3$</td>
<td>$n^3 \log n$</td>
<td>$2^n$</td>
<td>$(n^2 + 4)^{1/2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applying the theorem about order classes, and the fact that a sum of terms is $\Theta$ of its dominant term, we quickly get the following rough answer (except the highlighted ones):

$\Theta(1)$ 5000
$\Theta(\log \log n^5)$ $\log \log n^2$
$\Theta(\log n) \log n^2$
$\Theta((\log n)^5) (\log n)^5$
$\Theta(n^{0.3}) n^{0.3}$
$\Theta(n) n + \log n 4n + n^{1/2} (n^2 + 4)^{1/2}$
$\Theta(n^2) n^2 n^2 \log n$
$\Theta(2^n) 2^n$
$\Theta(3^n) 3^n$

That leaves the highlighted four unclassified. Making guesses and then applying the limit theorem to verify the answer can rank those.
For example, \((n^2 + 4)^{1/2}\) is \(\Theta(n)\) because:

\[
\lim_{n \to \infty} \frac{(n^2 + 4)^{1/2}}{n} = \lim_{n \to \infty} \left( \frac{n^2 + 4}{n^2} \right)^{1/2} = \lim_{n \to \infty} \left( 1 + \frac{4}{n^2} \right)^{1/2} = 1
\]

By using l'Hopital’s Rule, it’s easy to show that \(n^2 \log n\) is not \(O(n^2)\) or \(O(n^3)\) but that it lies between them.

Again using l’Hopital’s Rule, it’s easy to show that \(\log \log n^2\) is between 1 and \(\log n\) and that \((\log n)^5\) lies between \(\log n\) and \(n\).

4. Using any theorems from the course notes, give a formal proof that:

\[(n - 5)(n + \log n + \sqrt{n})\] is \(\Theta(n^2)\)

**Proof:** Multiplying out, the given function is:

\[f(n) = n^2 + n \log n + n^{3/2} - 5n - 5 \log n - 5n^{1/2}\]

By Theorem 5, each term of this function is \(O(n^2)\), so we might guess that the function is \(\Theta(n^3)\). To show that, work out the following limit and apply Theorem 8:

\[
\lim_{n \to \infty} \frac{f(n)}{n^2} = \lim_{n \to \infty} \frac{n^2 + n \log n + n^{3/2} - 5n - 5 \log n - 5n^{1/2}}{n^2} = \lim_{n \to \infty} \left( 1 + \frac{\log n}{n} + \frac{1}{n^{1/2}} - \frac{5}{n} - \frac{5 \log n}{n^2} - \frac{5}{n^{3/2}} \right) = 1
\]

5. Decide if each of the following statements is true or false — no justification is necessary.

(a) \((3 \log n)^3 - 10 \sqrt{n} + 2n\) is \(\Theta(n)\)

**TRUE**

The question is "how does the first term compare to the last one"? If you work out the limit of the two, you find that \(n / (\log n)^2\) goes to infinity as \(n\) does, so the dominant term is \(n\).

(b) \(\sqrt{n^2 - 10n + 100}\) is \(\Omega(n)\)

**TRUE**

Again, just take the limit as \(n\) goes to infinity of the ratio, which requires a little algebra… the limit turns out to equal 1, so the radical is \(\Theta(n)\) and by definition that makes it \(\Omega(n)\).
(c) \(2^n - n^3\) is \(\Omega(n^4)\) \hspace{1cm} \text{TRUE}

From a theorem, the dominant term here is \(2^n\), so this is \(\Theta(2^n)\). From the list of categories in another theorem, \(2^n\) is \(\Omega(\text{any constant power of } n)\).

6. Suppose that executing an algorithm on input of size \(N\) requires executing \(T(N) = N \log N + 8N\) instructions. How long would it take to execute this algorithm on hardware capable of carrying out \(2^{23}\) instructions per second if \(N = 2^{30}\)? (Give your answer in hours, minutes and seconds, to the nearest second.)

The total number of instructions that must be executed is \(T(2^{30})\). Using the given formula for \(T(N)\), we get:

\[
T(2^{30}) = 2^{30} \log(2^{30}) + 8 \cdot 2^{30} = 30 \cdot 2^{30} + 8 \cdot 2^{30} = 38 \cdot 2^{30}
\]

The number of seconds needed to execute that many instructions would be \(T(N)\) divided by the speed of the hardware, in instructions per second. So, we get:

\[
time = \frac{T(2^{30})}{2^{23}} = \frac{38 \cdot 2^{30}}{2^{23}} = 38 \cdot 2^7 = 4864
\]

That works out to be 1 hour, 21 minutes, 4 seconds (1:21:04) total running time.