You will submit your solution to this assignment to the Curator System (as `HW2`). Your solution must be either a plain text file (e.g., NotePad) or a typed MS Word document; submissions in other formats will not be graded.

Partial credit will only be given if you show relevant work.

---

1. [20 points] Apply the exact analysis rules from the course notes (Slide 7, T06.AlgorithmAnalysis.pdf) to determine a complexity function T(N) for the following algorithm. You must simplify your answer completely (no summation formulas, all terms combined as far as possible).

```
Limit = N;                       // 1
Result = 1;                      // 1

for (i = 1; i <= N; i++) {       // 1 before, 2 per pass; 1 to exit

   if ( i % 2 == 0 )             // 2 per pass
      Result = i * Result;       // 2, if done
   else
      Result = Result + i * i;   // 3, if done
}
```

$$T(N) = 3 + \sum_{i=1}^{N}\left(2 + 2 + \max(2,3)\right) + 1 = \sum_{i=1}^{N}(7) + 4 = 7N + 4$$

If you counted i++ as 2 operations, you would get: $8N + 4$.

2. [20 points] Apply the exact analysis rules from the course notes (Slide 7, T06.AlgorithmAnalysis.pdf) to determine a complexity function T(N) for the following algorithm. You must simplify your answer completely (no summation formulas, all terms combined as far as possible).

```
Limit = N;                       // 1
Sum = 1;                         // 1

for (i = 1; i <= N; i++) {       // 1 before, 2 per outer pass, 1 to exit

   for (j = 1; j <= i; j++) {    // 1 before, 2 per inner pass, 1 to exit

      Sum = Sum + i * j;         // 3
   }
}
```

$$T(N) = 3 + \sum_{i=1}^{N}\left(2 + 1 + \sum_{j=1}^{i}(2+3) + 1\right) + 1 = \sum_{i=1}^{N}\left(\sum_{j=1}^{i}(5) + 4\right) + 4 = \sum_{i=1}^{N}(5i + 4) + 4 = 5\frac{N(N+1)}{2} + 4N + 4$$

$$= \frac{5}{2}N^2 + \frac{13}{2}N + 4$$

If you counted i++ and j++ as 2 operations, you would get: $3N^2 + 8N + 4$.

3. [20 points] Apply the exact analysis rules from the course notes (Slide 7, T06.AlgorithmAnalysis.pdf) to determine a complexity function T(N) for the following algorithm. You must simplify your answer completely (no summation formulas, all terms combined as far as possible).

```
Limit = 1 << N;                        // Limit == 2^N:  2
Sum = 0;                               // 1
X = 1;                                 // 1

for (i = 1; i <= N; i = 2 * i) {  // 1 before, 3 per pass, 1 to exit

   Sum = Sum + i * X;                  // 3 per pass
   X++;                                // 1 per pass
}
```

The key here is that the loop body will be executed $1+\lceil \log N \rceil$ times since i will count from 2^0 to $N = 2^{\log N}$.

$$T(N) = 5 + \sum_{pass=1}^{1+\lceil \log N \rceil} (3+3+1) + 1 = \sum_{pass=1}^{1+\lceil \log N \rceil} (7) + 6 = 7\left(1+\lceil \log N \rceil\right) + 6 = 7\lceil \log N \rceil + 13$$

If you counted X++ as 2 operations, you would get: $8\lceil \log N \rceil + 14$.

4. [15 points] State the simplest possible big-Θ equivalent for each given function. No justification is required.

a) $f(n) = 3n\log n + 17n^2 + 8\log n + 42n + 100$

$f(n)$ is $\Theta\left(n^2\right)$

b) $g(n) = 1000n + \log n$

$g(n)$ is $\Theta(n)$

c) $h(n) = 10 + 7n^{50} + 2^n$

$h(n)$ is $\Theta\left(2^n\right)$

5. [25 points] Use any applicable theorems from the course notes to prove the following two facts:

a) $f(n) = n^2 \log n + n \log^2 n$ is $\Theta\left(n^2 \log n\right)$

b) $f(n) = n^2 \log n + n \log^2 n$ is strictly $\Omega\left(n \log^2 n\right)$

Note: $\log^2 n = \left(\log n\right)^2$.

The only theorems that will help with this are the Limit Theorem (Theorem 8) and its corollary. First of all:

$$\operatorname*{limit}_{n\to\infty} \frac{n^2 \log n + n \log^2 n}{n^2 \log n} = \operatorname*{limit}_{n\to\infty}\left(1 + \frac{\log n}{n}\right) = 1 + \operatorname*{limit}_{n\to\infty} \frac{1/n \ln 2}{1} = 1 + \operatorname*{limit}_{n\to\infty} \frac{1}{n \ln 2} = 1$$

Therefore, by Theorem 8, the first statement is true.

And:

$$\operatorname*{limit}_{n\to\infty} \frac{n^2 \log n + n \log^2 n}{n \log^2 n} = \operatorname*{limit}_{n\to\infty}\left(\frac{n}{\log n} + 1\right) = 1 + \operatorname*{limit}_{n\to\infty} \frac{1}{1/n \ln 2} = 1 + \operatorname*{limit}_{n\to\infty} n \ln 2 = \infty$$

So, by the Corollary to Theorem 8, the second statement is also true.