You will submit your solution to this assignment to the Curator System (as `HW03`). 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.

If you work with a partner, be sure to put both the name and PID of each partner at the beginning of the file you submit.

Except as noted, credit will only be given if you show relevant work.

In all questions about complexity, functions are assumed to be nonnegative.

1. [30 points] The analysis of a certain algorithm leads to the following complexity function (for the average case):

$$T(N) = 43N^2 \log N + 15N^2 + \log N + 2$$

Several computer science students offer their conclusions about the algorithm (quoted below). For each conclusion, state whether it is correct or incorrect, based on the given information, and give a brief justification of your answer; feel free to cite any relevant theorems from the course notes.

a) Jimmy Earl Neer of WVU says that the algorithm, on average, is $O(N^2)$.

b) Haskell Hoo V of UVA says no, the algorithm, on average, is $\Omega(N)$.

c) Jimmy Earl Neer retorts that the algorithm, on average, is actually $\Omega(N^2 \log N)$.

d) Joe Hokie of VT offers the suggestion that the algorithm, on average, is $\Theta(N \log N)$.

e) Haskell Hoo V asserts that, in the worst case, the algorithm must be $O(N^2 \log N)$.

f) Joe Hokie replies that, in the worst case, the algorithm could certainly be $\Theta(N^3)$.

g) Haskell Hoo V says that, in the worst case, the algorithm must be $\Omega(\log N)$.

h) Jimmy Earl Neer says that the algorithm's best case performance must be $\Omega(N^2 \log N)$.

i) Joe Hokie says, no, the algorithm's best case performance could be $\Omega(N \log N)$.

j) Haskell Hoo V then claims that the algorithm's best case performance must be $\Omega(\log N)$.

2. [24 points] Suppose that an algorithm takes 60 seconds for an input of 2^8 elements. Estimate how long the same algorithm, running on the same hardware, would take if the input contained 2^{16} elements, and that the algorithm's complexity function is:

a) $\Theta(N)$
b) $\Theta(\log N)$
c) $\Theta(N \log N)$
d) $\Theta(N^2)$

Assume that the low-order terms of the complexity functions are insignificant, and state your answers to the nearest tenth of a second. Be sure to show supporting work.

3. [24 points] Use theorems from the course notes to solve the following problems. Show work to support your conclusions.

a) Find the "simplest" function g(n) such that

$$f(n) = 17n^2 + 3n\log n + 1000 \text{ is } \Theta(\text{g(n)})$$

b) Find the "simplest" function g(n) such that

$$f(n) = 5n\log^2 n + 8n^2 \log \text{n is } \Theta(\text{g(n)})$$

c) Find the "simplest" function g(n) such that

$$f(n) = \sqrt{n} + \log\left(8\text{n}^2\right) \text{ is } \Theta(\text{g(n)})$$

4. [12 points] Using the counting rules from the course notes, find the exact-count complexity function T(n) for the following algorithm. Show details of your analysis, and simplify your answer.

```
for (r = 1; r <= 2*n; r++) {
   a[r][1] = 1;
   for (c = 2; c <= r; c += 2) {
      a[r][c] = c * a[r][c-1];
   }
}
```

5. [10 points] Use the definitions of O and Ω to prove that:

$$f(n) \text{ is } \mathrm{O}\big(g(n)\big) \text{ if and only if } g(n) \text{ is } \Omega\big(f(n)\big)$$