Please answer each of the following problems. Refer to the course webpage for the **collaboration policy,** as well as for **helpful advice** for how to write up your solutions.

1. **Fun with Big-O notation.** (6 points; 1 point each) Mark the following as `True` or `False`. Briefly but convincingly justify all of your answers, using the definitions of $O(\cdot)$, $\Theta(\cdot)$ and $\Omega(\cdot)$. [**To see the level of detail we are expecting, the first question has been worked out for you.**]

   (z) $n = \Omega(n^2)$. This statement is `False`. To see this, we will use a proof by contradiction. Suppose that, as per the definition of $\Omega(\cdot)$, there is some n_0 and some $c > 0$ so that for all $n \geq n_0$, $n \geq c \cdot n^2$. Choose $n = \max\{1/c, n_0\} + 1$. Then $n \geq n_0$, but we have $n > 1/c$, which implies that $c \cdot n^2 > n$. This is a contradiction.

   (a) $n = O(n \log(n))$.

   (b) $n^{1/\log(n)} = \Theta(1)$.

   (c) If
   $$f(n) = \begin{cases} 5^n & \text{if } n < 2^{1000} \\ 2^{1000}n^2 & \text{if } n \geq 2^{1000} \end{cases}$$
   and $g(n) = \frac{n^2}{2^{1000}}$, then $f(n) = O(g(n))$.

   (d) For all possible functions $f(n), g(n) \geq 0$, if $f(n) = O(g(n))$, then $2^{f(n)} = O(2^{g(n)})$.

   (e) $5^{\log\log(n)} = O(\log(n)^2)$

   (f) $n = \Theta\left(100^{\log(n)}\right)$

2. **n-naught not needed.** (3 points) Suppose that $T(n) = O(n^d)$, and that $T(n)$ is never equal to ∞. Prove rigorously that there exists a c so that $0 \leq T(n) \leq c \cdot n^d$ for all $n \geq 1$. That is, the definition of $O(\cdot)$ holds with $n_0 = 1$. [**We are expecting a rigorous proof using the definition of $O(\cdot)$**].

3. **Fun with recurrences.** (6 points; 1 point each)

   Solve the following recurrence relations; i.e. express each one as $T(n) = O(f(n))$ for the tightest possible function $f(n)$, and give a short justification. Be aware that some parts might be slightly more involved than others. Unless otherwise stated, assume $T(1) = 1$. [**To see the level of detail expected, we have worked out the first one for you.**]

   (z) $T(n) = 6T(n/6) + 1$. We apply the master theorem with $a = b = 6$ and with $d = 0$. We have $a > b^d$, and so the running time is $O(n^{\log_6(6)}) = O(n)$.

   (a) $T(n) = 2T(n/2) + 3n$

   (b) $T(n) = 3T(n/4) + \sqrt{n}$

   (c) $T(n) = 7T(n/2) + \Theta(n^3)$

(d) $T(n) = 4T(n/2) + n^2 \log n$

(e) $T(n) = 2T(n/3) + n^c$, where $c \geq 1$ is a constant (that is, it doesn't depend on n).

(f) $T(n) = 2T(\sqrt{n}) + 1$, where $T(2) = 1$

4. **Different-sized sub-problems.** (6 points) Solve the following recurrence relation.

$$T(n) = T(n/2) + T(n/4) + T(n/8) + n,$$

where $T(1) = 1$. **[We are expecting a formal proof. You may state your final running time with $O(\cdot)$ notation, but do not use it in your proof.]**

5. **What's wrong with this proof?** (9 points) Consider the following recurrence relation:

$$T(n) = T(n-5) + 10 \cdot n$$

for $n \geq 5$, where $T(0) = T(1) = T(2) = T(3) = T(4) = 1$. Consider the following three arguments.

1. **Claim: $T(n) = O(n)$.** To see this, we will use strong induction. The inductive hypothesis is that $T(k) = O(k)$ for all $5 \leq k < n$. For the base case, we see $T(5) = T(0) + 10 \cdot 5 = 51 = O(1)$. For the inductive step, assume that the inductive hypothesis holds for all $k < n$. Then

$$T(n) = T(n-5) + 10n,$$

and by induction $T(n-5) = O(n-5)$, so

$$T(n) = O(n-5) + 10n = O(n).$$

This establishes the inductive hypothesis for n. Finally, we conclude that $T(n) = O(n)$ for all n.

2. **Claim: $T(n) = O(n)$.** To see this, we will use the Master Method. We have $T(n) = a \cdot T(n/b) + O(n^d)$, for $a = d = 1$ and

$$b = \frac{1}{1 - 5/n}.$$

Then we have that $a < b^d$ (since $1 < 1/(1-5/n)$ for all $n > 0$), and the master theorem says that this takes time $O(n^d) = O(n)$.

3. **Claim: $T(n) = O(n^2)$.** Imagine the recursion tree for this problem. (Notice that it's not really a "tree," since the degree is 1). At the top level we have a single problem of size n. At the second level we have a single problem of size $n-5$. At the t'th level we have a single problem of size $n - 5t$, and this continues for at most $t = \lfloor n/5 \rfloor + 1$ levels. At the t'th level for $t \leq \lfloor n/5 \rfloor$, the amount of work done is $10(n-5t)$. At the last level the amount of work is at most 1. Thus the total amount of work done is at most

$$1 + \sum_{t=0}^{\lfloor n/5 \rfloor} 10(n - 5t) = O(n^2).$$

(a) (3 points) Which, if any, of these arguments are correct? [**We are expecting a single sentence stating which are correct.**]

(b) (6 points) For each argument that you said was incorrect, explain why it is incorrect. If you said that all three were incorrect, then give a correct argument. [**We are expecting a few sentences of detailed reasoning for each incorrect algorithm; and if you give your own proof we are expecting something with the level of detail of the proofs above—except it should be correct!**]