NAME:

1. In each part, show the algebraic steps that prove that the given function solves the given recurrence relation. Be sure to show a reasonable number of steps.

   (a) Show that $f(n) = n^2 \lg(n)$ is an exact solution of $T(n) = 4\,T(n/2) + n^2$.

   Solution:

$$\begin{aligned}
\text{plug } f(n) \text{ into rhs of recurrence} &= ??? && (1)\\
&= ??? && (2)\\
&= ??? && (3)\\
&= ??? && (4)\\
&= n^2 \lg(n) && (5)
\end{aligned}$$

   (b) Show that $f(n) = \lg(n!) + 1$ is an exact solution of $T(n) = T(n-1) + \lg(n)$.

   Solution:

$$\begin{aligned}
\text{plug } f(n) \text{ into rhs of recurrence} &= ??? && (6)\\
&= ??? && (7)\\
&= ??? && (8)\\
&= ??? && (9)\\
&= \lg(n!) + 1 && (10)
\end{aligned}$$

   (c) Show that $f(n) = \lg(n)\ \lg(\lg(n))$ is an exact solution of $T(n) = 2\,T(\sqrt{n}) + \lg(n)$.

   Solution:

$$\begin{aligned}
\text{plug } f(n) \text{ into rhs of recurrence} &= ??? && (11)\\
&= ??? && (12)\\
&= ??? && (13)\\
&= ??? && (14)\\
&= \lg(n)\ \lg(\lg(n)) && (15)
\end{aligned}$$

2. In each part below, use the Master Theorem to determine an asymptotic bound (in Θ-notation) for the solution of the recurrence relation. Briefly explain each of your answers.

(a) $T(n) = 2\,T(n/3) + 1$.

Solution:

(b) $T(n) = 5\,T(n/4) + n$.

Solution:

(c) $T(n) = 2\,T(n/4) + \sqrt{n}$.

Solution:

(d) $T(n) = 7\,T(n/7) + n$.

Solution:

3. Suppose you are choosing between the following three algorithms.

   - Algorithm A solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
   - Algorithm B solves problems of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.
   - Algorithm C solves problems of size n by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $\Theta(n^2)$ time.

   What are the running times of each of these algorithms (in big-Θ notation), and which should you choose?

   Solution:

4. Let $M(n)$ denote the number of multiplications needed to compute a^n, where a and n are positive integers.

   (a) Write pseudo code for a recursive, divide and conquer algorithm that computes a^n and has a recurrence relation for $M(n)$ given by

   $$M(n) = M(n-1) + 1$$

   Solution:

```
power(a,n)
{


}
```

(b) Write pseudo code for a recursive, divide and conquer algorithm that computes a^n and has a recurrence relation for $M(n)$ given by

$$M(n) = \begin{cases} M(n/2) + 1, & \text{if } n \text{ is even} \\ M(n/2) + 2, & \text{if } n \text{ is odd} \end{cases}$$

Solution:

```
power(a,n)
{


}
```

(c) What are the running times for these two algorithms (in big-Θ notation)?

Solution: