The Master Method and its use

The Master method is a general method for solving (getting a closed form solution to) recurrence relations that arise frequently in divide and conquer algorithms, which have the following form:

$$T(n) = aT(n/b) + f(n)$$

where $a \geq 1, b > 1$ are constants, and $f(n)$ is function of non-negative integer n. There are three cases.

(a) If $f(n) = O(n^{\log_b a-\epsilon})$, for some $\epsilon > 0$, then $T(n) = \Theta(n^{log_b a})$.

(b) If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$.

(c) If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some $\epsilon > 0$, and $af(n/b) \leq cf(n)$, for some $c < 1$ and for all n greater than some value n', Then $T(n) = \Theta(f(n))$.

For some illustrative examples, consider
(a) $T(n) = 4T(n/2) + n$
(b) $T(n) = 4T(n/2) + n^2$
(c) $T(n) = 4T(n/2) + n^3$

In these problems, $a = 4, b = 2$, and $f(n) = n, n^2, n^3$ respectively. We compare $f(n)$ with $n^{\log_b a} = n^{\log_2 4}$. The three recurrences satisfy the three different cases of Master theorem.
(a) $f(n) = n = O(n^{2-\epsilon}$ for, say, $\epsilon = 0.5$. Thus, $T(n) = \Theta(n^{log_b a}) = \Theta(n^2)$.
(b) $f(n) = n^2 = \Theta(n^2)$, thus $T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n^2 \log n)$.
(c) $f(n) = n^3 = \Omega(n^{2+\epsilon}$ for, say, $\epsilon = 0.5$ and $af(n/b) \leq cf(n)$, i.e., $4(\frac{n}{2})^3 = \frac{n^3}{2} \leq cn^3$ for $c = 1/2$. Thus, $T(n) = \Theta(f(n)) = \Theta(n^3)$.

(d) The recurrence for binary search is $T(n) = T(n/2) + \Theta(1)$. Using Master Theorem, $a = 1, b = 2, f(n) = \Theta(1)$. Now $f(n) = \Theta(1) = \Theta(n^{\log_b a}) = \Theta(n^0) = \Theta(1)$. Using the second form of Master Theorem, $T(n) = \Theta(n^0 \log n) = \Theta(\log n)$.

(e) $T(n) = 4T(n/2) + n^2 \log n$. This does not form any of the three cases of Master Theorem straight away. But we can come up with an upper and lower bound based on Master Theorem.

Clearly $T(n) \geq 4T(n) + n^2$ and $T(n) \leq 4T(n) + n^{2+\epsilon}$ for some $\epsilon > 0$. The first recurrence, using the second form of Master theorem gives us a lower bound of $\Theta(n^2 \log n)$. The scond recurrence gives us an upper bound of $\Theta(n^{2+\epsilon})$. The actual bound is not clear from Master theorem. We use a recurrence tree to bound the recurrence.

$$
\begin{aligned}
T(n) &= 4T(n/2) + n^2 \log n \\
&= 16T(n/4) + 4(\frac{n}{2})^2 \log n/2 + n^2 \log n \\
&= 16T(n/4) + n^2 \log n/2 + n^2 \log n \\
&= \ldots
\end{aligned}
$$

$$
\begin{aligned}
T(n) &= n^2 \log n + n^2 \log n/2 + n^2 \log n/4 + \ldots + n^2 \log n/(2^{\log n}) \\
&= n^2(\log n + \log n/2 + \log n + 4 + \ldots) \\
&= n^2(\log n \cdot n/2 \cdot n/4 + \ldots + n/(2^{logn})) \quad \text{(Transforming logs)} \\
&= n^2(\log 2^{\log n}) \quad \text{(Using geometric series)} \\
&= n^2 \log n \qquad \text{(Using } 2^{logn} = n\text{)}
\end{aligned}
$$

Thus, $T(n) = n^2 \log n$.