Question 1

(a) $T(n) = 3T(n/4) + \Theta(n\log n)$

$a=3$, $b=4$, $f(n) = \Theta(n\log n)$

$n^{\log_b a} = n^{\log_4 3} \approx n^{0.792}$

since $n\log n$ faster than $n^{0.792}$

$af(\frac{n}{b}) = 3(\frac{n}{4}\log\frac{n}{4}) = \frac{3}{4}n(\log n - \log 4) < cn\log n$

for constant $c<1$ and large $n$, use case 3

$T(n) = \Theta(f(n)) = \Theta(n\log n)$

(b) $T(n) = 5T(n/2) + n^2\log n$

$a=5$, $b=2$, $f(n) = \Theta(n^2\log n)$

$n^{\log_b a} = n^{\log_2 5} \approx n^{2.32}$ vs $n^2\log n$

Since $f(n) = \Theta(n^2\log n)$ has an extra $\log n$ factor compared to $n^{\log_2 5}$, use Case 2

$T(n) = \Theta(n^{2.32})$

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

$a=4$, $b=2$, $f(n) = \Theta(n^2\log n)$

$\log_2 4 = 2$, so $n^{\log_2 4} = n^2$

$f(n) = \Theta(n^2 \log n)$ , $n^{\log_2 4} = \Theta(n^2)$

Since $f(n)$ has an extra $\log n$ term compared to $n^2$,

use case 2 with $k=1$

$T(n) = \Theta(n^2 \log^2 n)$

Question 2

(a) Each recursive call reduce n by 2,

level i the size should be $\frac{n}{2^i}$, stop when size $=1$

which is $\frac{n}{2^L} = 1$

so $L = \log_2 n$

(b) At Level i, $2^i$ subproblems,

so the number of leaf nodes is $2^L = 2^{\log_2 n} = n$

(c) Each leaf node is $T(1)$, which is assumed to be a $\Theta(1)$

Since n leaf nodes, total cost is $n \cdot \Theta(1) = \Theta(n)$

(d) Level i, there are $2^i$ sub, size is $\frac{n}{2^i}$.

so cost for each Level is $2^i \times \frac{n}{2^i} = n$

And there are $\log_2 n$ levels, so the total cost is $\Theta(n \log n)$