Question 1:

(a) Can't tell which algorithm is faster, since big O notation only tells us the upper bound each algorithm.

(b) R1 is faster. Algorithm 1 takes linear time at most while Algorithm 2 takes at least quadratic time.

(c) Can't tell which algorithm is faster, Since R2 could run in $\theta(1)$ time, which us faster than R1 or it could run in $\theta(n^2)$ time which is slower than R1.

(d) R1 runs faster. Algorithm 1 takes linear time at most while Algorithm 2 takes exactly quadratic time.

Question 2:

$f(n) = 3n^2 + 10n + 729$

$$\begin{aligned}(a) f(n) &\leq 3n^2 + 10n + 729n \quad \text{ when } n \geq 729 \\ &= 3n^2 + 739n \\ &\leq 3n^2 + 739n^2 \quad \text{ when } n \geq 739 \\ &= 742n^2 = O(n^2)\end{aligned}$$

$$\begin{aligned}(b) f(n) &\leq 742n^2 \quad \text{proved in part(a)} \\ &\leq 742n^3 \quad \text{ when } n \geq 742 \\ &= O(n^3)\end{aligned}$$

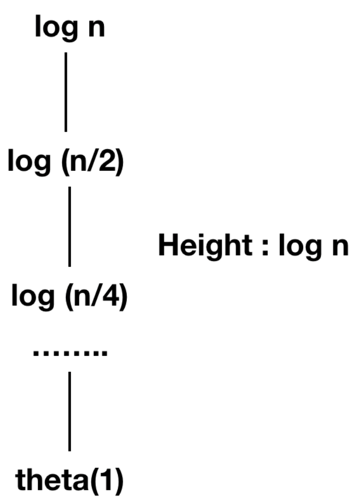
$$\begin{aligned}(c) f(n) &\geq 3n^2 + 10n \\ &\geq 10n \quad \text{ since } n^2 \geq 0 \\ &= \Omega(n)\end{aligned}$$

$$\begin{aligned}(c) f(n) &\geq 3n^2 + 10n \\ &\geq 3n^2 \quad \text{ when } n \geq 0 \\ &= \Omega(n^2)\end{aligned}$$

(d) $O(n^2)$ and $\Omega(n^2)$ are better bounds.
This is because they have a smaller range. That is to say, every function which is $O(n^2)$ is $O(n^3)$ but not vice versus.
For example, $3n^3$ is $O(n^3)$ but not $O(n^2)$

Question 3:

(a)



Level sum shrinks and the max level sum is at root
$H(t) \leq height \times \max(\text{level sum}) = log(n) \times log(n) = O(log^2(n))$

(b)
Guess: Lower bound is $O(log^2(n))$
Base case: $H(2) = \Theta(1) = d > clog^2(2) = c \iff 0 < c < d$
Induction hypothesis: $H(n) > clog^2(n)$ for all $n < k$
Induction step:

$$
\begin{aligned}
H(n) &= H(\frac{n}{2}) + \log(n) \\
&\geq clog^2(\frac{n}{2}) + log(n) \\
&= c(log(n) - log(2))^2 + log(n) \\
&= c(log^2(n) - 2log(n)log(2) + log^2(2)) + log(n) \\
&= clog^2(n) + clog^2(2) + log(n) \times (1 - 2clog(2)) \\
&\geq clog^2(n) \text{ when } 1 - 2clog(2) > 0 \iff 0 < c < \frac{1}{2log(2)} \\
&= \Omega(log^2(n))
\end{aligned}
$$

(c)
$a = 1, b = 2, \quad f(n) = \log n$
$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$ so $f(n) = \Theta(n^{\log_b a} \log n)$
Applying master method, we know that, $H(n) = \Theta(f(n) \log n) = \Theta(\log^2 n)$

(d)
The bounds match.
I would prefer using master method on this question, since it can solve the upper bound and lower bound simultaneously in just a few lines.

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + .. + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n})$$

$R_T = 0$ for all T except terminal state

$$Q_{t+n}(S_t) = Q_{t+n-1}(S_t) + \alpha[G_{t:t+n} - Q_{t+n-1}(S_t)]$$