1)

a) $w(n) = 2w(n/3) + 1$

$\log_b(a) = \log_3(2)$

$1 = n^0$

$n^0 < n^{\log_3(2)}$ so $T(n) = \Theta(n^{\log_3(2)})$

b) $w(n) = 5w(n/4) + n$

$\log_b(a) = \log_4(5)$

$n^1 < n^{\log_4(5)}$ so $T(n) = \Theta(n^{\log_4(5)})$

c) $w(n) = 7w(n/7) + n$

$\log_b(a) = \log_7(7) = 1$

$n^1 = n^1$ so $T(n) = \Theta(n \log n)$

d) $w(n) = 9w(n/3) + n^2$

$\log_b(a) = \log_3(9) = 2$

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

e) $w(n) = 8w(n/2) + n^3$

$\log_b(a) = \log_2(8) = 3$

$n^3 = n^3$ so $T(n) = \Theta(n^3 \log n)$

f) $w(n) = 49w(n/25) + n^{3/2} \log n$

$\log_b(a) = \log_{25}(49)$

$n^{3/2} \log n > n^{\log_{25}(49)}$ so $T(n) = \Theta(n^{3/2} \log n)$

g) $w(n) = w(n-1) + 2$

$w(n-1) = w(n-2) + 2$

$w(n-2) = w(n-3) + 2$

this has a constant linear relationship of $2n \in O(n)$

h) $w(n) = w(n-1) + n^c$ where $c \geq 1$

$w(n-1) = w(n-2) + (n-1)^c$

$w(n-2) = w(n-3) + (n-2)^c$

as we go down we are left with $n^c + (n-1)^c + (n-2)^c$

as n is sufficiently large we are then left with

$n(n^c) = n^{c+1} \in O(n^c)$

I) $w(n) = w(\sqrt{n}) + 1$

$= w(n^{1/4}) + 1 + 1$

$= w(n^{1/8}) + 1 + 1 + 1$

The equation for a given level is $w(n^{1/2^k}) + k$

if we set $n^{1/2^k} = 2$ then

$n = 2^{2^k}$

$\log_2 n = 2^k$

$\log_2 \log_2 n = k$ so $T(n) = \Theta(\log\log(n))$

2)

A. $w(n) = 5w(n/2) + n$

$\log_b(a) = \log_2(5)$

$n^1 < n^{\log_2(5)}$ so $T(n) = \Theta(n^{\log_2(5)})$

B. $w(n) = 2(n-1) + 1$

it will take k times to reach $w(0)$ and level has 2^k 1's added to it.

$2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^k = 2^{k+1} - 1$. Assuming $n = k$, $2^{n+1} - 1 \in O(2^n)$

C. $w(n) = 9w(n/3) + O(n^2)$

$\log_b(a) = \log_3(9) = 2$

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

I would choose algorithm C as it has the smallest runtime as $n \to \infty$