Q2-1

$i=0)$ $T(n) = 9T(\frac{n}{3}) + n$

~~$\Rightarrow 9(9T(\frac{n}{9}) + \frac{n}{3}) + n = 9^2T(\frac{n}{9}) + 3n$~~

~~$\Rightarrow 9^2(9T(\frac{n}{27}) + \frac{n}{3}) + n = 9^3T(\frac{n}{27}) + \dots$~~

$i=1)$ $9(9T(\frac{n}{9}) + \frac{n}{3}) + n = 9^2T(\frac{n}{9}) + 3n + n$

$i=2)$ $9(9^2T(\frac{n}{27}) + n + \frac{n}{3}) + n = 9^3T(\frac{n}{27}) + 9n + 3n + n$

$9^k T(\frac{n}{3^k}) + \underbrace{\sum_{i=0}^{k-1} 9^i \frac{n}{3^i}}_{a\frac{1-r^k}{1-r}}$ ~~$= 9^k T(\frac{n}{3^k}) + \frac{n}{3}$~~

base case: $\frac{n}{3^k} = 1 \Rightarrow k = \log_3(n)$

$T(n) = 9^k T(\frac{n}{3^k}) + a\frac{1-r^k}{1-r}$

$= 9^k T(\frac{n}{3^k}) + n\frac{1-3^k}{1-3}$

$= 9^{\log_3(n)} T(\frac{n}{3^{\log_3(n)}}) + n\frac{1-3^{\log_3(n)}}{-2}$

$= 9^{\log_3(n)} T(\frac{n}{n}) + \frac{n-n^2}{-2}$

$O(9^{\log_3(n)} T(1) + \frac{n-n^2}{-2})$

$\boxed{O(n^2)}$

$T(n) \leq an^2$ Assuming $T(n) = O(n^2)$

Base: $n=1$

$T(n) = 9T(\frac{n}{3}) + n = 9[a(\frac{n}{3})^2] + n$

$T(1) = 9[a(\frac{1}{3})^2] + 1 = O(1) = O(n^2) = O(1)$ ✓

$n = k+1$

$T(n) = 9[a(\frac{n}{3})^2] + n = 9[a(\frac{k+1}{3})^2] + k+1$

$O(9[a\underbrace{(\frac{k+1}{3})^2}_{\text{dominating term}}] + k+1 = O(k^2)$ ✓

Q2-2)

			$n=1{,}000{,}000$
10^n	$n=1$; 10	$n=1000$; $10^{1000} = 1\times10^{1000}$	$1\times10^{1{,}000{,}000}$
n^n	$n=1$; 1	$n=1000$; $1{,}000^{1000} = 1\times10^{3000}$	$1\times10^{6{,}000{,}000}$
$\log_2 n$	$n=1$; 0	$n=1000$; $\log_2(1000) = \sim10$	~20
$2\sqrt{\log_2 n}$	$n=1$; 0	$n=1000$; $2\sqrt{\log_2 1000} = \sim6$	~9
$\sqrt{2n}$	$n=1$; $\sqrt{2}$	$n=1000$; $\sqrt{2000} = \sim45$	~1400
$n^{2.5}$	$n=1$; 1	$n=1000$; $1000^{2.5} = \sim3\times10^7$	$\sim1\times10^{15}$

To check growth, $n=1$, $n=1000$, + $n=1{,}000{,}000$ were inputted into the equations and the outputs were observed. The more the output increases with increasing n, the more growth.

smallest growth rate ⟵⟶ largest growth rate

$$\boxed{2\sqrt{\log_2(n)},\ \log_2(n),\ \sqrt{2n},\ n^{2.5},\ 10^n,\ n^n}$$

Q3-1

a) $\log_2 n = k$, time $= O(n^2)$, so each sublist takes $O(k^2)$

\# of sublists $= \frac{n}{k}$, so Total Time $= \frac{n}{k} \times O(k^2)$
$= \boxed{O(nk)}$

b) $\log_2 n = k$, \# sublists $= \frac{n}{k}$, so work at each level $= O(n)$

\# of levels $= O(\log_2(\frac{n}{k}))$, so Total Time $= \boxed{O(n\log_2(\frac{n}{k}))}$

c) $O(n\log_2(n)) = O(nk + n\log_2(\frac{n}{k}))$

$nk + n\log_2(\frac{n}{k})$
$= nk + n\log_2(n) - n\log_2(k)$

If $k = \log_2(n)$:
$n\log_2(n) + n\log_2(n) - n\log_2(\log_2(n))$
$= 2n\log_2(n) - n\log_2(\log_2(n))$

$O(2n\log_2(n) - n\log_2(\log_2(n)))$
$= O(2n\log_2(n))$
$\boxed{= O(n\log_2(n))}$ ✓ $\boxed{k \leq \log_2(n)}$