1:

Order by Growth Increasing Rate:
$2/N, 12, \sqrt{N}, N, N \log\log N,\ N \log N, N \log(N^2), N \log^2 N, N^{1.5},\ N^2, N^2 \log N, N^3, 2^{N/2}, 2^N$

Table of comparisons on final page, where the tabular values denote the results of a limit comparison test (x denotnig infinities). Only one pair of identical growth was found, which makes sense by the rules of logarithms.

2:

$Given : a_n = a_{n-1}^2, a_0 = 2, a_1 = 4, a_2 = 16$

a) Alice's fee for hour N:
Looking at the pattern, $2^2 = 4, 2^4 = 16, 2^8 = 256$, it is noted that the degree of 2 is represented by 2^N every step.
$\boxed{Fee = 2^{2^N}}$

b) Estimate for number of hours to reach S Hourly:
Using the formula for part a, then algebraically solving for N.
$S(N) = 2^{2^N}, N(S) = \log_2(\log_2 S), \boxed{O(log_2(log_2 S))}$

c) Estimate for number of hours to reach S Total:
$S(N)_{total} = \Sigma_{N=1}^{K} 2^{2^N}$, so $N(S)$ is $O(\log_2(\log_2 S)), N$ times, bringing it to $O(N \log_2(\log_2 S)) = \boxed{O((\log_2(\log_2 S))^{(\log_2(\log_2 S))})}$.

3:

a) $O(N)$
$y = cx, y(100p) = 0.8ms, c = \frac{0.8ms}{100p}$
$y(x) = 1.20E5ms, x = \frac{100*1.20E5p}{0.8} = \boxed{1.50E7p}$

b) $O(N \log N)$
$y = cx \log x, y(100p) = 0.8ms, c = 1.74E - 3\frac{ms}{p},$

$x\log x(y) = y/c, x\log x(1.2E5ms) \rightarrow \boxed{4.51E6p}$

c) $O(N^2)$
$y = cx^2, y(100p) = 0.8ms, x(y) = \sqrt{\frac{y}{c}},$
$c = 8E-5\frac{ms}{p}, x(1.20E5)^2 \rightarrow \boxed{38729p}$

d) $O(N^3)$
$x(y) = (\frac{y}{c})^{\frac{1}{3}}, c = 8E-7\frac{ms}{p},$
$x(1.20E5) \rightarrow \boxed{5313p}$

e) $O(2^N)$
$x(y) = \log_2\frac{y}{c}, c = 6.31E-31\frac{ms}{p},$
$x(1.20E5) \rightarrow \boxed{81p}$

0.1 4:

$Given : Foo(N) = 221N\log_2 N, Bar(N) = 2N^2$

It is noted that the two functions share solutions in the following way.

$$\frac{2}{221} = \frac{\log_2 N}{N} = \log_2 N^{\frac{1}{N}} \rightarrow 2^{\frac{2}{221}} = N^{\frac{1}{N}}$$

Considering that it is impossible to complete any decimal of an operation;
For $N < 1, Bar > Foo$. For $1 < N < 1120, Bar < Foo$. For $N > 1120, Bar > Foo$.

a) $N > 10,000$
Using the ranges previously established.
Better runtime: $\boxed{Foo}$

b) $N < 100$
Better runtime: $\boxed{Bar}$

c) $N = 1000$
Number of operations is just under the cross over point.
Using $\frac{1}{\Delta N}\int_{N_1}^{N_2} f(N)dN =< f(N) >$ ranging from 0 to 1000
we find $< Foo >= 1E6, < Bar >= 6.7E5$
Better avg runtime: $\boxed{Bar}$

d) Will Bar always be faster than Foo?
No, Foo completes the first operation instantanously, and as mentioned, Foo will be the faster algorithm for operation sets (N) greater than 1120.

	Numerator													
Denominator	2/N	12	\sqrt{N}	N	N \log \log N	N \log N	N \log(N^2)	N \log^2 N	N^{1.5}	N^2	N^2 \log N	N^3	2^{N/2}	2^N
2/N	1	x	x	x	x	x	x	x	x	x	x	x	x	x
12	0	1	x	x	x	x	x	x	x	x	x	x	x	x
\sqrt{N}	0	0	1	x	x	x	x	x	x	x	x	x	x	x
N	0	0	0	1	x	x	x	x	x	x	x	x	x	x
N \log \log N	0	0	0	0	1	x	x	x	x	x	x	x	x	x
N \log N	0	0	0	0	0	1	2	x	x	x	x	x	x	x
N \log(N^2)	0	0	0	0	0	1/2	1	x	x	x	x	x	x	x
N \log^2 N	0	0	0	0	0	0	0	1	x	x	x	x	x	x
N^{1.5}	0	0	0	0	0	0	0	0	1	x	x	x	x	x
N^2	0	0	0	0	0	0	0	0	0	1	x	x	x	x
N^2 \log N	0	0	0	0	0	0	0	0	0	0	1	x	x	x
N^3	0	0	0	0	0	0	0	0	0	0	0	1	x	x
2^{N/2}	0	0	0	0	0	0	0	0	0	0	0	0	1	x
2^N	0	0	0	0	0	0	0	0	0	0	0	0	0	1