1. (a) By the master method, \( a = 2, b = 2, \) and \( f(n) = 5 \leq n^{\log_b a} = n^{\log_a b} \), so \( t_n \leq Kn^{\log_b a} = Kn \) for some constant \( K \).

(b) \( a = 2, b = 4, \) and \( f(n) = n \). So \( f(n) \geq n^{\log_b a} = n^{\log_4 2} \), and \( t_n \leq Kf(n) = Kn \).

(c) \( a = 1, b = 2, \) and \( f(n) = 2n-1 \). In this case, \( \log_b a = \log_2 1 = 0 \), so \( t_n \) must be at most proportional to \( f(n) \):

\[
t_n \leq K(2n-1)
\]

(d) \( a = 3, b = 3, \) and \( f(n) = 4 \). So \( t_n \leq Kn^{\log_b a} = Kn \).

(e) \( a = 16, b = 2, \) and \( f(n) = 5n \). \( n^{\log_b a} = n^4 \) and \( f(n) \leq 5n^4 \), so \( t_n \leq Kn^4 \).

(f) \( a = 4, b = 2, \) and \( f(n) = 3n \). \( n^{\log_b a} = n^2 \) and \( f(n) = 3n \leq n^2 \), so \( t_n \leq Kn^2 \).

2. (a) To find the largest number in a set, we can recursively break the set in half, find the max of each half, and then return the max of these maxes. More formally:

\[
\text{FIND-MAX}(x_1, \ldots, x_n)
\]

\[
\begin{align*}
&\text{if } n = 1 \\
&\quad \text{then return } x_1 \\
&\text{if } n = 2 \\
&\quad \text{then return max}(x_1, x_2) \\
&\quad m_1 = \text{FIND-MAX}(x_1, \ldots, x_{n/2}) \\
&\quad m_2 = \text{FIND-MAX}(x_{n/2+1}, \ldots, x_n) \\
&\quad \text{return max}(m_1, m_2)
\end{align*}
\]

(b) The number of comparisons performed by this algorithm depends on the number of inputs. If \( n = 1 \), then no comparisons are performed: \( c_1 = 0 \). If \( n = 2 \), then one comparison is performed: \( c_2 = 1 \). Otherwise, we recurse twice on half of the set, and perform one final comparison: \( c_n = 2c_{n/2} + 1 \).

(c) Using the master method, we have \( a = 2, b = 2, \) and \( f(n) = 1 \). So \( f(n) < n \) and \( t_n \) is proportional to \( n^{\log_b a} = n \).