1. Assume your machine has 64 bit words. Assume you can multiply two $n$ word numbers in time $3n^2$ with a standard algorithm. Assume you can multiply two $n$ word numbers in time $12n^{\log_3 3}$ with a “fancy” algorithm.

(a) Approximately, how large does $n$ have to be for the fancy algorithm to be better?
(b) How many bits is that?
(c) How many decimal digits is that?

2. Use the same assumptions as for problem (1), except assume that your “fancy” algorithm is twice as fast so that you can multiply two $n$ word numbers in time only $6n^{\log_3 3}$.

(a) Approximately, how large does $n$ have to be for the fancy algorithm to be better?
(b) How many bits is that?
(c) How many decimal digits is that?

3. Cascade Merge thought of as a recursive algorithm as follows: Assume there are $n$ sorted lists, $S[1], S[2], \ldots, S[n]$, each of size $m$. Recursively Cascade Merge the first $n-1$ lists and then Merge the last list (of size $m$) into the sorted list $A$ (of size $(n-1)m)$.

(a) Write down the recursive version of Cascade Merge in psuedocode. It can be very high level. You can assume that you have a merge routine available.
(b) Derive a recurrence for the exact number of comparisons the algorithm uses.
(c) Guess the solution to your recurrence.
(d) Prove your solution is correct using Mathematical Induction.

4. Consider the following recurrence, defined for $n$ a power of 4:

$$T(n) = \begin{cases} 
6 & \text{if } n = 1 \\
3T(n/4) + 5n + 2 & \text{otherwise}
\end{cases}$$

Solve the recurrence exactly using the “Master Theorem” (below).

“Master Theorem”

$$T(n) = \begin{cases} 
aT(n/b) + cn^d & n > 1 \\
f & n = 1
\end{cases}$$

implies

$$T(n) = \begin{cases} 
\left( f + \frac{c}{ab^{d-1}} \right) n^{\log_b a} - \frac{cn^d}{ab^{d-1}} = \begin{cases} 
\Theta(n^{\log_b a}) & a > b^d \\
\Theta(n^d) & a < b^d
\end{cases} & a = b^d
\end{cases}.$$

Summing solutions: If

$$T(n) = \begin{cases} 
aT(n/b) + \sum c_i n^{d_i} & n > 1 \\
f & n = 1
\end{cases}$$

then we can just sum the solutions of each recurrence:

$$T_i(n) = \begin{cases} 
aT_i(n/b) + c_i n^{d_i} & n > 1 \\
0 & n = 1
\end{cases}$$

and add in $fn^{\log_b a}$ for the contribution from the leaves.