Put your *name* and *section number* on your solution.

1. Assume your machine has 32 bit words. Assume you can multiply two $n$ word numbers in time $2n^2$ with a standard algorithm. Assume you can multiply two $n$ word numbers in time $10n^\log_2 3$ 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 you can multiply two $n$ word numbers in time only $5n^{\log_2 3}$ with a “fancy” algorithm.
   (b) How many bits is that?
   (c) How many decimal digits is that?

3. Selection Sort can be thought of as a recursive algorithm as follows: Find the largest element and put it at the end of the list (to be sorted). Recursively sort the remaining elements.
   (a) Write down the recursive version of Selection Sort in pseudocode.
   (b) Derive a recurrence for the exact number of comparisons the algorithm uses.
   (c) Use the iteration method to solve the recurrence. Simplify as much as possible.
   (d) Use mathematical induction to verify your solution.

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

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

   (a) Solve the recurrence exactly using the iteration method. Simplify as much as possible.
   (b) 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

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

   Summing solutions: If

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

   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.