1. Assume that each word of your machine has 64 bits. Assume that you can multiply two n-word numbers in time $5n^2$ with a standard algorithm. Assume that you can multiply two n-word numbers in time $10n^{\lg 3}$ with a "fancy" algorithm. For each part *briefly justify* and *show your work*.

   (a) Approximately, how large does n have to be for the fancy algorithm to be better?

   (b) Approximately how many bits is that?

   (c) Approximately how many decimal digits is that?

2. Assume that you multiply the two two-digit numbers 36 and 52 using the method that does only three atomic multiplies. Show that steps of the algorithm on this example.

3. Consider the following recurrence (for the time of some algorithm).

$$T(n) = 5T(n/3) + 4n + 1, \qquad T(1) = 2 \ .$$

   (a) Calculate $T(9)$ by hand. Show your work.

   (b) Use the tree method to solve the recurrence exactly, assuming n is a power of 3.

4. We can multiply large integers recursively by splitting each integer into thirds (rather than halves as done in class). In order to multiply the two three-digit numbers abc and def the standard algorithm would do nine atomic multiplications.

   (a) Explain how you can do fewer atomic multiplications by forming the product $(a+b+c)(d+e+f)$. How many atomic multipications do you use?

   (b) Write a recurrence for how fast this version of integer multiplication is. In order to keep this simple, just use $a\alpha n$, where a is a constant, for the time to do all of the additions at each invocation of the routine. Assume n is a power of three.