CMSC 203 – Extra Credit HW 1
Due Thursday 4/2

Kevin Winner

March 11, 2013

All coding tasks may be done in your language of choice. All functions should be written recursively for full credit, although partial credit will be given for an iterative implementation. While I recommend you complete these tasks in the order I have prescribed, you are welcome to pick and choose which parts you would like to complete.

This assignment will essentially be graded as a point return on your lowest homework grade. This assignment will be graded out of 100 points, and any points earned will be added to the score of your lowest homework, to a maximum of 100 points on that assignment. Each of the 5 parts is worth 20 points.

Successfully completing all parts of this project will require you to apply algorithm analysis, proof by induction, recursion, and number theory. It will be hard. Quite hard. That's why it's extra credit.

- Encode the Sieve of Eratosthenes algorithm described on page 259 of your textbook. Your implementation should be a recursive function which takes two parameters: the list of integers to operate on and the current smallest prime (start at 2) for which to remove multiples from the list. You'll also need a wrapper function to initialize the appropriate list of integers.
- What is the Big-Oh complexity of your Sieve of Eratosthenes implementation in terms of k, the maximum value in your list of integers?
- Prove, using a proof by strong induction, that any integer $n \geq 1$ is either prime or can be written as the product of primes.
- Write a recursive function to find the prime factorization of an integer n. I strongly recommend using your implementation of the Sieve of Eratostenes to identify all the primes $\leq \frac{n}{2}$, then use this list as part of your factorization algorithm.
- Use your prime factorization method to complete Project Euler Problem #3, which asks "what is the largest prime factor of the number 600851475143?" This problem will likely require your implementation to be highly efficient, and may require optimization beyond that required for the earlier problems.