Complexity for Probabilists
Homework 1

1. Show that the following functions are primitive recursive[1]:

   (a) $f_1 : \mathbf{N} \to \mathbf{N}$ given by $f_1(x) = x!$.

   (b) $f_2 : (\mathbf{N} \times \mathbf{N}) \to \mathbf{N}$ given by $f_2(x, y) = \max(x, y)$.

   (c) $f_3 : \mathbf{N} \to \mathbf{N}$, where f_3 is any function which is "eventually 0", i.e. which has the property that there exists an $n \in \mathbf{N}$ such that $f_3(x) = 0$ for all $x \geq n$.

2. Suppose that the infinite set $A \subseteq \mathbf{N}$ is recursively enumerable. Does this imply that there exists a computer program which on input $x \in \{1, 2, \ldots\}$ outputs the x:th smallest element of A?

3. **Dead code.** It is of practical interest to eliminate dead code (i.e. instructions that are never executed) from computer programs. Does there exist a program which identifies dead code? By this I mean a program which takes as input (M, x), where M is a computer program (in your favorite language!) and $x \in \mathbf{N}$, and outputs

$$\begin{cases} 1 & \text{if there exists a } y \in \mathbf{N} \text{ such that the } x\text{:th line of the program } M \\ & \text{is ever executed on input } y, \\ 0 & \text{otherwise.} \end{cases}$$

4. **Goldbach's conjecture**[2]**.** Is the following function computable (in the sense of the Church–Turing thesis)?

$$f(x) = \begin{cases} 1 & \text{if Goldbach's conjecture is true,} \\ 0 & \text{otherwise,} \end{cases}$$

   for every $x \in \mathbf{N}$.

[1] For one or more of these, it may be useful to build upon the construction of some primitive recursive functions (Add, Mult,...) on p 6-8 of Håstad: Complexity Theory.

[2] Goldbach's conjecture says that every even number ≥ 6 can be written as the sum of two primes. It is a famous open problem to tell whether this is true or false.