Q1. (4 points) Big O

What does the statement $f(x)$ is $O(x^2)$ mean, precisely. Either give a mathematical definition or an English-language definition that is correct and complete.

Q2. (4 points) Algorithms

What is wrong (if anything) with the following algorithm for factoring a number into prime numbers? Assume it is to be used by a grade school math student or a common computer. Do not worry about the language constructs, just the core expression of the algorithm:

To factor the number $n$ into prime factors:
   let $r$ be an empty list;
   append $n$ to the list $r$
   append 1 to the list $r$
   * if $n$ is 1, return $r$, otherwise:
     let $f$ = the largest prime factor of $n$
     add $f$ onto list $r$
     let $n$ = $n$ / $f$
     goto the line marked *
Q3. (12 points) Algorithmic Complexity.

For each part of this problem, consider the task of looking up the phone number of a friend, where you have a set of (name, phone number) pairs for your friends. You may assume that you have \( n \) friends in your set.

In Big-O terms, what is the computational complexity of looking up the phone number by linearly searching through every friend until you find them? (3 points)

Suppose you invest \( O(n \log n) \) time to sort your set (don't worry about how), and can access the \( k^{th} \) item in the set in \( O(1) \) time (i.e., they are available for random access). Now what is the time-complexity of looking up a phone number using the most efficient mechanism (some form of binary search)? (3 points)

You're trying to decide whether it is worth taking the time to sort your list of friends. Assume you expect to look up \( p \) phone numbers before you add new friends. What is the total complexity of looking up \( p \) phone numbers using linear and binary search, including the time needed to sort the numbers for binary search? When is it worth sorting them? (4 points)

Bonus problem: Suppose you had a hash table in which you could store your list. What is the time-complexity of looking up a phone number in a hash table (this is from class, not the text)? (2 points)