To whom it may concern:

Software patents are a horrid idea: they hurt individuals, in that they take away the rights of individuals to intelligent devices they have, and cast a pall of uncertainty upon all software developers, as to whether their code infringes upon an existing patent.

Many similar concepts in software arise from independent sources when solving a problem, because it is often the case that simple logic dictates a certain way of doing something. A patented algorithm is usually not un-obvious, but the logical conclusion of work towards a goal. Many, many algorithms comprise a large software package such as a word processor or operating system. Looking back at previous source code for projects reveals that things are done over and over again sometimes nearly identically.

The concept of software patents is toxic. It makes any software endeavor a possible patent mine, and disagreements over a piece of code being infringing is far harder than determining a physical object that may infringe. What is the standard for determining uniqueness--

Lines of code that are different?

Code that is different from the patented item, but which performs a complex task with identical results?

What of the case of possible infringement when the challenged code is written in a radically different language?

What standard can be set for the size or complexity of a patentable item?

Given the length of the computer industry, the length of patents is an absurdly long time. What happened five years ago is a long time, yet patents last several times that. With "patent trolls" on the rise, virtually all software projects are at the mercy of interests whose primary interest is money. This has the side effect of stifling innovation, from those without the resources to withstand a legal barrage.

I do not believe the Supreme Court has ever ruled on this concept. Software patents are not American--they are fueled by greed. The USPTO should not grant these patents.

Stephan F. Andre'
Ann Arbor, Michigan
[e-mail redacted]