Discussion Questions for "Measuring Performance"

1. Using Big-Oh notation, if we declare that an algorithm that takes 100,000,000 instructions to execute, no matter how big the input data size is, and call it O(1), and then compare that to another algorithm which takes 12 instructions for each input, so if the input size is n, it takes 12n instructions, and so is O(n), then we say the O(1) algorithm is "much faster" than the O(n) algorithm. However, for any practical input size data, the O(n) algorithm executes a lot fewer instructions than the O(1) algorithm. Is this fair? Does this example shed any light on the accuracy of the Big-Oh notation?

2. If I have an algorithm that, for n inputs, executes $c_2 n^2 + c_1 n + c_0$ instructions where all the c's are constants, what is the order of that algorithm?

3. I knew someone in college who invented a sort algorithm than ran faster than O(n log n). Why isn't he famous?

4. Why do academics have a different attitude about scalability than industry professionals? What can be done to sensitize academics about scalability issues?