We begin our study of algorithms with a motivating example and an overview of the use of the scientific method for studying algorithm performance.

Lecture: Union−Find. We illustrate our basic approach to developing and analyzing algorithms by considering the dynamic connectivity problem. We introduce the *union–find* data type and consider several implementations (quick find, quick union, weighted quick union, and weighted quick union with path compression). Finally, we apply the union–find data type to the percolation problem from physical chemistry.

Lecture: Analysis of Algorithms. The basis of our approach for analyzing the performance of algorithms is the scientific method. We begin by performing computational experiments to measure the running times of our programs. We use these measurements to develop hypotheses about performance. Next, we create mathematical models to explain their behavior. Finally, we consider analyzing the memory usage of our Java programs.

Programming Assignment: Percolation. Your programming assignment will give you an opportunity to apply these concepts to a fundamental problem in physical chemistry. It is the first of many examples where a good algorithm—in this case, weighted quick union—makes the difference between being able to efficiently solve a problem and not being able to address it at all.

Job Interview Questions. Algorithmic interview questions based on the lecture material.

Suggested Readings. Section 1.4 and 1.5 in *Algorithms, 4th edition*.