coursera

Overview

Our lectures this week are centered on the idea of problem-solving models like maxflow and shortest path, where a new problem can be formulated as an instance of one of those problems, and then solved with a classic and efficient algorithm. To complete the course, we describe the classic unsolved problem from theoretical computer science that is centered on the concept of algorithm efficiency and guides us in the search for efficient solutions to difficult problems.

Lecture 13: Reductions. In this lecture our goal is to develop ways to classify problems according to their computational requirements. We introduce the concept of reduction as a technique for studying the relationship among problems. People use reductions to design algorithms, establish lower bounds, and classify problems in terms of their computational requirements.

Lecture 14: Linear Programming (optional). The quintessential problem-solving model is known as linear programming, and the simplex method for solving it is one of the most widely used algorithms. In this lecture, we given an overview of this central topic in operations research and describe its relationship to algorithms that we have considered.

Lecture 15: Intractability. Is there a universal problem-solving model to which all problems that we would like to solve reduce and for which we know an efficient algorithm? You may be surprised to learn that we do no know the answer to this question. In this lecture we introduce the complexity classes P, NP, and NP-complete, pose the famous P = NP question, and consider implications in the context of algorithms that we have treated in this course.

To Do:

- **Job Interview Questions.** Algorithmic interview questions based on the lecture material.
- **Suggested Readings.** Section 6.5—6.6 (pp. 903-921) in *Algorithms, 4th edition*.

Mark as completed