Week Ten Reading Notes

In the early days of computing, the question was, “what can and cannot be computed”. After lots of success, the question then shifted to, “what can be computed in a reasonable amount of time”. To study this question, computer scientists identified a bunch of computational problems that were easy to describe but whose algorithms initially seemed to take a long time.

After intensive studies, many PhD thesis, and novel mathematics, computer scientists noticed that many of these problems can be formulated in similar ways with similar efficient algorithmic solutions. One goal of this course is to recognize these forms so that if your problem, whatever it may be, across any number of disciplines, fits these forms, then you will know how to solve them.

There are two extreme ways to teach this material. One way is to describe all the problems and then all the solutions, being as general as possible. The other way is to present one at a time: the problem and its solution, and then hope the student gets the general approach.

There are pluses and minuses to each of these teaching approaches. This week's lectures are a combination of approaches. Lecture Sequence 18 begins with an overview of the range of problems and their descriptions. Then, one at a time problem/solution approach is followed. In other words, do not get overwhelmed with the solution.

The optimization lectures could be called "graph problems”. Graphs are a very efficient and universal way of representing a large number of problems. Anything that has connections, physical or logical, can be represented as a graph or a tree. A few examples of graphs might be: a family ancestor tree, a friendship graph (think connections between people who have “friended” one another on Facebook), a connection between lectures and exam questions, and many, many more. It is safe to say that you will see graphs everywhere after these lectures.

One last bit of advice: always remember, in computer science, trees grow upside down!

-Larry Rudolph