Overview

We begin our exploration of advanced algorithms by looking at data structures for graphs, which are essential models for a broad variety of real-world problems. A number of classic algorithms are easy to implement, but worthy of careful study both because convincing oneself that they operate as intended can be a challenge and because their performance is strongly dependent on proper use of the basic data structures that we covered in Algorithms Part I.

Lecture 1: Undirected Graphs. We define an undirected graph API and consider the adjacency-matrix and adjacency-lists representations. We introduce two classic algorithms for searching a graph depth-first-search and breadth-first-search. We also consider the problem of computing connected components and conclude with related problems and applications.

Lecture 2: Directed Graphs. In this lecture we study directed graphs. We begin with depth-first search and breadth-first search in digraphs and describe applications ranging from garbage collection to web crawling. Next, we introduce a depth-first search based algorithm for computing the topological order of an acrylic digraph. Finally, we implement the Kosaraju-Sharir algorithm for computing the strong components of a digraph.

To do:

- **Exercises**. Drill exercises on the lecture material.
- **Programming Assignment: WordNet**. Determine the semantic relatedness of two nouns using the WordNet lexicon.
- **Job Interview Question**. Algorithmic interview questions based on the lecture material.
- **Suggested Readings.** Section 4.1 and 4.2 in Algorithms, 4th edition.