Graph Theory

Representation, Storage Alternatives, Traversing, Searching, Operations, Graph Drawing, Project

A graph is one of the most versatile structures in mathematics with widespread research & practical uses. We see use of graph concepts in areas ranging from the knowledge graph, the social graph and in organization-specific graphs (e.g. Microsoft Graph). Devs are used to graph terms such as the "object graph" or the "call graph". Graph databases are becoming very popular. Anywhere you see the work "network" in industry there is a mathematical graph lurking underneath. Because of its popularity, we even see use of the "graph" term where it shouldn't be – e.g. GraphQL (which is neither graph-based, nor indeed, a query language).

This course covers all aspects of graph theory, from simple representation, to traversal and search, to transformations, to display. Starting with simple edges and vertices (with storage as either a matrix or a linked list [preferred for sparsely populated graphs]), we investigate how to build, transform and present graphs.

We explore how to efficiently handle large graphs with a keen interest in high performance. This course also covers graph drawing (a surprising complex topic in its own right). We conclude with a project to build a graph engine.

Target Audience

This course is aimed mathematicians and developers who wish to become familiar with the theoretical and practical usage of graphs in a variety of scenarios.

Prerequisites

Attendees need a good foundation in mathematics and programming, as this course will be covering graph-related ideas from both disciplines.

Attendees need to be familiar with one programming language. Any will do, as in the hands-on labs they will be developing a graph library in that language from first principles.

Contents of One-Day Training Course

Overview of Graph Theory

Part of discrete mathematics
No "top" or root
Overview of graph concepts
Vertices and edges

Introduction to Graphs

G = (V, E)
where V is a node set and E is an edge set
Simple graph
Multi-graph
Common operations on graphs
Basic terminology
Directed (digraphs) vs. undirected graphs

Traversing A Graph

The visitor pattern
Ensuring each node is visited
Weighted graph (e.g. assign a "cost" to each edge, to influence selected path)

Searching A Graph

Breath-first search
Depth-first search
Dealing with cycles

Graph Operations

Graph composition: merge points to be based on identity of edges
Subsumption
Node selection
Edge contraction

Types of Graphs

Connected graph
Bipartite graph
Complete graph
Regular graph

Graph Properties

Shortest path
Minimum spanning trees
Traveling salesman

A Tree As A Kind Of Graph

Every tree is a graph, but not reverse
Binary search trees
Subtrees

Graph As Network Flow

Networks appear throughout engineering, science, business and daily life – how to best represent as graphs
Max flow / min cut theorem
Interacting with networks as graphs

Graph Data Structures

Adjacency matrix vs. adjacency list
How to optimize for large graphs
Indexing
Improving storage on hard disk

Graph Databases

Review of how graphs are handled by a graph database
Querying possibilities
Sample usage

Graph Drawing

How to automate placing graph constructs on a planar surface
The crossing number
How to improve automated layout

Project

Creating a custom graph engine to store and interact with large-scale graphs efficiently