



K. Erciyes

Algebraic Graph Algorithms

A Practical Guide Using Python

There has been unprecedented growth in the study of graphs, which are discrete structures that have many real-world applications. The design and analysis of algebraic algorithms to solve graph problems have many advantages, such as implementing results from matrix algebra and using the already available matrix code for sequential and parallel processing.

Providing Python programming language code for nearly all algorithms, this accessible textbook focuses on practical algebraic graph algorithms using results from matrix algebra rather than algebraic study of graphs. Given the vast theory behind the algebraic nature of graphs, the book strives for an accessible, middle-ground approach by reviewing main algebraic results that are useful in designing *practical* graph algorithms on the one hand, yet mostly using graph matrices to solve the graph problems. Python is selected for its simplicity, efficiency and rich library routines; and with the code herein, brevity is forsaken for clarity.

Topics and features:

- Represents graphs by algebraic structures, enabling new, robust methods for algorithm analysis and design
- Provides matroid-based solutions to some graph problems, including greedy algorithm problems
- Offers Python code that can be tested and modified for various inputs
- Supplies practical hints, where possible, for parallel processing associated with algebraic algorithms
- Links to a web page with supportive materials

This clearly arranged textbook will be highly suitable for upper-level undergraduate students of computer science, electrical and electronic engineering, bioinformatics, and any researcher or person with background in discrete mathematics, basic graph theory and algorithms.

Dr. Kayhan Erciyes is a full Professor in the Department of Software Engineering at Maltepe University, Istanbul, Turkey. His other publications include the Springer titles *Discrete Mathematics and Graph Theory*, *Distributed Real-Time Systems*, *Guide to Graph Algorithms*, *Distributed and Sequential Algorithms for Bioinformatics*, and *Distributed Graph Algorithms for Computer Networks*.

ISBN 978-3-030-87885-6



9 783030 878856



1	Introduction	1
1.1	Graphs	1
1.2	Matrices	2
1.3	Algebraic Graph Algorithms	2
1.4	Python Language	3
1.5	Challenges	4
1.6	Outline of the Book	5
1.6.1	Background	5
1.6.2	Basic Graph Algorithms	5
	References	7
 Part I Background		
2	A Short Review of Python	11
2.1	Introduction	11
2.2	Data Structures	12
2.2.1	Strings	13
2.2.2	Lists	13
2.2.3	Tuples	15
2.2.4	Arrays	16
2.2.5	Dictionaries	18
2.2.6	Sets	19
2.3	Flow Control	19
2.3.1	The <i>if</i> Statement	19
2.3.2	Loops	20
2.4	Functions	22
2.5	Modules	24
2.6	Chapter Notes	25
	Reference	26
3	Basic Matrix Computations	27
3.1	Introduction	27
3.2	Matrix Operations	28

3.3	Transpose of a Matrix	30
3.4	Matrix Properties	31
3.5	Types of Matrices	31
3.5.1	Diagonal Matrix	31
3.5.2	Identity Matrix	32
3.5.3	Triangular Matrix	32
3.6	Matrix Multiplication	33
3.6.1	Vector-Vector Multiplication	33
3.6.2	Matrix-Vector Multiplication	34
3.6.3	Matrix-Matrix Multiplication	35
3.6.4	Boolean Matrix Multiplication	37
3.7	Determinant of a Matrix	37
3.8	Matrix Inverse	39
3.9	Rank Computation	39
3.9.1	Powers of a Matrix	40
3.10	Eigenvalues and Eigenvectors	42
3.11	Block Matrices	45
3.12	Chapter Notes	45
	Reference	46
4	Graphs, Matrices and Matroids	47
4.1	Introduction to Graphs	47
4.1.1	Degree of a Vertex	48
4.1.2	Subgraphs	49
4.2	Graph Operations	49
4.3	Types of Graphs	51
4.3.1	Digraph	52
4.3.2	Complete Graphs	52
4.3.3	Weighted Graphs	53
4.3.4	Bipartite Graphs	53
4.3.5	Regular Graphs	53
4.3.6	Graph Isomorphism	54
4.4	Walks, Paths, Cycles	54
4.5	Graphs and Matrices	56
4.5.1	Adjacency Matrix	56
4.5.2	Adjacency List	57
4.5.3	Incidence Matrix	57
4.5.4	Cycle Matrix	58
4.5.5	Path Matrix	59
4.5.6	The Laplacian	60
4.6	Python Graphs	61
4.7	Matroids	63
4.7.1	The Basic Matroid	64

4.7.2	The Graphic Matroid	65
4.7.3	Weighted Matroid	65
4.7.4	Graphs, Matroids and Matrices	66
4.8	Chapter Notes	66
	References	68
5	Parallel and Sparse Matrix Computations	69
5.1	Parallel Processing	69
5.1.1	Architectures	69
5.1.2	Basic Communications	70
5.1.3	Performance	70
5.1.4	Parallel Matrix Computations	71
5.2	Parallel Computations with Python	73
5.2.1	Process-Based Model with <i>Multiprocessing</i>	73
5.2.2	Message Passing Interface <i>mpi4py</i>	77
5.3	Sparse Matrix Algorithms	81
5.3.1	Sparsity Calculation	82
5.3.2	Sparse Matrix Representations	82
5.3.3	Sparse Matrix Multiplication	84
5.4	Chapter Notes	85
	References	86
 Part II Graph Algorithms		
6	Trees	89
6.1	Introduction	89
6.2	Spanning Tree Construction	90
6.3	Minimum Spanning Trees	93
6.3.1	Jarnik-Prim Algorithm	93
6.3.2	Kruskal's Algorithm as a Matroid	101
6.3.3	Boruvka's Algorithm	103
6.4	Chapter Notes	105
	References	107
7	Shortest Paths	109
7.1	Introduction	109
7.2	Breadth-First-Search	110
7.2.1	The Classical Algorithm	110
7.2.2	The Algebraic Algorithm	112
7.3	Dijkstra's Algorithm	114
7.4	Bellman-Ford Algorithm	117
7.5	Floyd-Warshall Algorithm	120

7.6	Transitive Closure	122
7.7	Chapter Notes	124
	References	125
8	Connectivity and Matching	127
8.1	Connectivity	127
8.1.1	Theory	127
8.1.2	Undirected Graph Connectivity	129
8.1.3	Directed Graph Connectivity	133
8.2	Matching	136
8.2.1	Unweighted Matching	137
8.2.2	Weighted Matching	138
8.2.3	Rabin-Vazirani Algorithm	144
8.3	Chapter Notes	145
9	Subgraph Search	149
9.1	Independent Sets	149
9.1.1	A Greedy Algorithm	149
9.1.2	Lowest Degree First Algorithm	151
9.1.3	Luby's Algorithm	153
9.2	Dominating Sets	155
9.3	Vertex Cover	157
9.3.1	A Greedy Algorithm	157
9.3.2	A 2-Approximation Algorithm	160
9.4	Coloring	162
9.4.1	Vertex Coloring	162
9.4.2	Edge Coloring	166
9.5	Chapter Notes	168
	References	170
10	Large Graph Analysis	171
10.1	Introduction	171
10.2	Degree Distribution	172
10.3	Density	173
10.4	Clustering Coefficient	176
10.5	Matching Index	178
10.6	Centrality	180
10.6.1	Degree Centrality	180
10.6.2	Closeness Centrality	182
10.6.3	Vertex Betweenness Centrality	184
10.6.4	Edge Betweenness Centrality	186
10.6.5	Eigenvalue Centrality	188

10.7	Network Models	190
10.7.1	Random Networks	190
10.7.2	Small-World Networks	191
10.7.3	Scale-Free Networks	193
10.8	Kronecker Graphs	194
10.9	Chapter Notes	197
	References	198
11	Graph Partitioning and Clustering	199
11.1	Graph Partitioning	199
11.1.1	BFS-Based Partitioning	200
11.1.2	Spectral Bisection	201
11.1.3	Multilevel Graph Partitioning	204
11.2	Graph Clustering	205
11.2.1	MST-Based Clustering	206
11.2.2	Shared Nearest Neighbor Clustering	209
11.2.3	Edge Betweenness Clustering	212
11.2.4	k -Core Clustering	214
11.3	Chapter Notes	215
	References	218
	Index	219