



# graphbook

A book on algorithmic graph theory

 [Project Home](#)[Downloads](#)[Wiki](#)[Issues](#)[Source](#)

**READ-ONLY:** This project has been [archived](#). For more information see [this post](#).

[Summary](#) [People](#)

## Project Information

[Project feeds](#)

### Code license

[GNU GPL v2](#)

### Labels

[Sage](#), [graphtheory](#),  
[algorithms](#)

### Members

[mvngu.n...@gmail.com](mailto:mvngu.n...@gmail.com),  
[wdjoy...@gmail.com](mailto:wdjoy...@gmail.com),  
[dphil...@usna.edu](mailto:dphil...@usna.edu),  
[dphi...@gmail.com](mailto:dphi...@gmail.com)

## Featured

### Downloads

[latest-r1991.pdf](#)  
[latest-r1991.tar.bz2](#)  
[latest-r1991.zip](#)  
[Show all »](#)

A [GNU-FDL](#) book on algorithmic graph theory by [David Joyner](#), [Minh Van Nguyen](#), and [David Phillips](#). This is an introductory book on algorithmic graph theory. Theory and algorithms are illustrated using the [Sage](#) open source mathematics software. To get an overview of the book, you can view the table of contents as shown below or download the complete book. This book is more commonly known as the "DaMNeD" book if you notice how our names are used to abbreviate the book. So feel free to call it the DaMNeD book :-)

Supplementary materials for the book can be found at  
<https://bitbucket.org/mvngu/graphbook-supplement>.

A companion book is [Explorations in Algebraic Graph Theory with Sage](#). This book covers how to use techniques from algebra to analyze many types of graphs.

The book is a work in progress; it is by no means complete yet. We as the authors work on it during our spare time. If you want to help out by contributing materials, that would be awesome and greatly appreciated.

**The latest revision of the book is called latest-rxxx.**

1. Introduction to graph theory
  1. Graphs and digraphs
  2. Subgraphs and other graph types
  3. Representing graphs in a computer
  4. Graph transformation
  5. Isomorphic graphs
  6. New graphs from old
2. Trees and forests
  1. Properties of trees
  2. Minimum spanning trees
  3. Binary trees
  4. Huffman codes
  5. Tree traversals
3. Shortest paths algorithms
  1. Distance labels
  2. Searching graphs
  3. Bellman-Ford algorithm
  4. Dijkstra's algorithm
  5. Topological sort
  6. All-pairs shortest paths
4. Graph data structures
  1. Priority queues
  2. Binary heaps
  3. Binomial heaps
  4. Binary search trees
5. Distance and connectivity
  1. Path and distance
  2. Vertex and edge connectivity
  3. Menger's theorem
  4. Network reliability
6. Centrality and prestige
  1. Vertex centrality
  2. Edge centrality

- 3. Ranking web pages
- 4. Hub and authority
- 7. Optimal graph traversals
  - 1. Eulerian graphs
  - 2. Hamiltonian graphs
  - 3. Chinese postman problem
  - 4. Traveling salesman problem
- 8. Graph coloring
  - 1. Vertex coloring
  - 2. Edge coloring
  - 3. Chromatic polynomial
  - 4. Assignment and scheduling
- 9. Maximum flow problems
  - 1. Flows and cuts
  - 2. Ford-Fulkerson theorem
  - 3. Edmonds-Karp algorithm
  - 4. Goldberg-Tarjan algorithm
  - 5. Path-flow decomposition
  - 6. Maximum weight closure
- 10. Algebraic graph theory
  - 1. Laplacian and adjacency matrices
  - 2. Eigenvalues and eigenvectors
  - 3. Algebraic connectivity
  - 4. Graph invariants
  - 5. Cycle and cut spaces
- 11. Random networks
  - 1. Network statistics
  - 2. Binomial random networks
  - 3. Erdos-Renyi networks
  - 4. Small-world networks
  - 5. Scale-free networks

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)