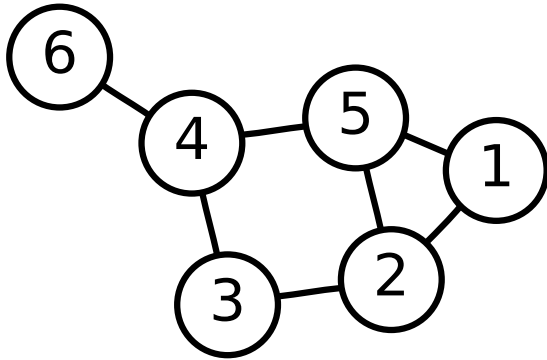
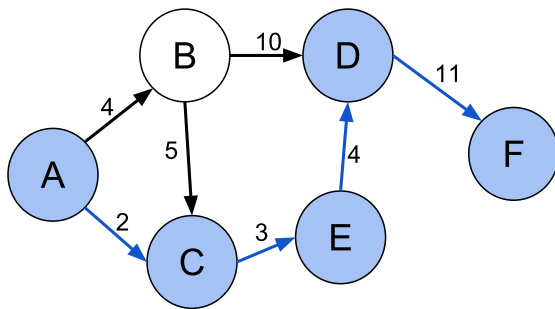


Shortest path problem



$(6, 4, 5, 1)$ and $(6, 4, 3, 2, 1)$ are both paths between vertices 6 and 1



Shortest path (A, C, E, D, F) between vertices A and F in the weighted directed graph

In **graph theory**, the **shortest path problem** is the problem of finding a **path** between two **vertices** (or nodes) in a **graph** such that the sum of the **weights** of its constituent edges is minimized.

The problem of finding the shortest path between two intersections on a road map (the graph's vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of its road segment) may be modeled by a special case of the shortest path problem in graphs.

1 Definition

The shortest path problem can be defined for **graphs** whether **undirected**, **directed**, or **mixed**. It is defined here for undirected graphs; for directed graphs the definition of path requires that consecutive vertices be connected by an appropriate directed edge.

Two vertices are adjacent when they are both incident

to a common edge. A **path** in an undirected graph is a **sequence** of vertices $P = (v_1, v_2, \dots, v_n) \in V \times V \times \dots \times V$ such that v_i is adjacent to v_{i+1} for $1 \leq i < n$. Such a path P is called a path of length $n - 1$ from v_1 to v_n . (The v_i are variables; their numbering here relates to their position in the sequence and needs not to relate to any canonical labeling of the vertices.)

Let $e_{i,j}$ be the edge incident to both v_i and v_j . Given a **real-valued** weight function $f : E \rightarrow \mathbb{R}$, and an undirected (simple) graph G , the shortest path from v to v' is the path $P = (v_1, v_2, \dots, v_n)$ (where $v_1 = v$ and $v_n = v'$) that over all possible n minimizes the sum $\sum_{i=1}^{n-1} f(e_{i,i+1})$. When each edge in the graph has unit weight or $f : E \rightarrow \{1\}$, this is equivalent to finding the path with fewest edges.

The problem is also sometimes called the **single-pair shortest path problem**, to distinguish it from the following variations:

- The **single-source shortest path problem**, in which we have to find shortest paths from a source vertex v to all other vertices in the graph.
- The **single-destination shortest path problem**, in which we have to find shortest paths from all vertices in the directed graph to a single destination vertex v . This can be reduced to the single-source shortest path problem by reversing the arcs in the directed graph.
- The **all-pairs shortest path problem**, in which we have to find shortest paths between every pair of vertices v, v' in the graph.

These generalizations have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest path algorithm on all relevant pairs of vertices.

2 Algorithms

The most important algorithms for solving this problem are:

- **Dijkstra's algorithm** solves the single-source shortest path problem.
- **Bellman–Ford algorithm** solves the single-source problem if edge weights may be negative.

- **A* search algorithm** solves for single pair shortest path using heuristics to try to speed up the search.
- **Floyd–Warshall algorithm** solves all pairs shortest paths.
- **Johnson’s algorithm** solves all pairs shortest paths, and may be faster than Floyd–Warshall on sparse graphs.
- **Viterbi algorithm** solves the shortest stochastic path problem with an additional probabilistic weight on each node.

Additional algorithms and associated evaluations may be found in Cherkassky, Goldberg & Radzik (1996).

3 Single-source shortest paths

3.1 Undirected graphs

3.2 Unweighted graphs

3.3 Directed acyclic graphs

An algorithm using **topological sorting** can solve the single-source shortest path problem in linear time, $\Theta(E + V)$, in weighted DAGs.

3.4 Directed graphs with nonnegative weights

The following table is taken from Schrijver (2004). A green background indicates an asymptotically best bound in the table.

This list is **incomplete**; you can help by **expanding it**.

3.5 Planar directed graphs with nonnegative weights

3.6 Directed graphs with arbitrary weights without negative cycles

This list is **incomplete**; you can help by **expanding it**.

3.7 Planar directed graphs with arbitrary weights

4 All-pairs shortest paths

The all-pairs shortest path problem finds the shortest paths between every pair of vertices v, v' in the graph.

The all-pairs shortest paths problem for unweighted directed graphs was introduced by Shimmel (1953), who observed that it could be solved by a linear number of matrix multiplications that takes a total time of $O(V^4)$.

4.1 Undirected graph

4.2 Directed graph

5 Applications

Shortest path algorithms are applied to automatically find directions between physical locations, such as driving directions on **web mapping** websites like **MapQuest** or **Google Maps**. For this application fast specialized algorithms are available.^[1]

If one represents a nondeterministic **abstract machine** as a graph where vertices describe states and edges describe possible transitions, shortest path algorithms can be used to find an optimal sequence of choices to reach a certain goal state, or to establish lower bounds on the time needed to reach a given state. For example, if vertices represent the states of a puzzle like a **Rubik’s Cube** and each directed edge corresponds to a single move or turn, shortest path algorithms can be used to find a solution that uses the minimum possible number of moves.

In a networking or telecommunications mindset, this shortest path problem is sometimes called the min-delay path problem and usually tied with a **widest path problem**. For example, the algorithm may seek the shortest (min-delay) widest path, or widest shortest (min-delay) path.

A more lighthearted application is the games of "**six degrees of separation**" that try to find the shortest path in graphs like movie stars appearing in the same film.

Other applications, often studied in **operations research**, include plant and facility layout, **robotics**, **transportation**, and **VLSI design**".^[2]

5.1 Road networks

A road network can be considered as a graph with positive weights. The nodes represent road junctions and each edge of the graph is associated with a road segment between two junctions. The weight of an edge may correspond to the length of the associated road segment, the time needed to traverse the segment, or the cost of traversing the segment. Using directed edges it is also possible to model one-way streets. Such graphs are special in the sense that some edges are more important than others for long distance travel (e.g. highways). This property has been formalized using the notion of highway dimension.^[3] There are a great number of algorithms that exploit this property and are therefore able to compute the shortest path a lot quicker than would be possible on

general graphs.

All of these algorithms work in two phases. In the first phase, the graph is preprocessed without knowing the source or target node. The second phase is the query phase. In this phase, source and target node are known. The idea is that the road network is static, so the preprocessing phase can be done once and used for a large number of queries on the same road network.

The algorithm with the fastest known query time is called hub labeling and is able to compute shortest path on the road networks of Europe or the USA in a fraction of a microsecond.^[4] Other techniques that have been used are:

- ALT (A* search, landmarks, and triangle inequality)
- Arc flags
- Contraction hierarchies
- Transit node routing
- Reach-based pruning
- Labeling

6 Related problems

For shortest path problems in computational geometry, see Euclidean shortest path.

The travelling salesman problem is the problem of finding the shortest path that goes through every vertex exactly once, and returns to the start. Unlike the shortest path problem, which can be solved in polynomial time in graphs without negative cycles, the travelling salesman problem is NP-complete and, as such, is believed not to be efficiently solvable for large sets of data (see P = NP problem). The problem of finding the longest path in a graph is also NP-complete.

The Canadian traveller problem and the stochastic shortest path problem are generalizations where either the graph isn't completely known to the mover, changes over time, or where actions (traversals) are probabilistic.

The shortest multiple disconnected path^[5] is a representation of the primitive path network within the framework of Reptation theory.

The widest path problem seeks a path so that the minimum label of any edge is as large as possible.

6.1 Strategic shortest-paths

Sometimes, the edges in a graph have personalities: each edge has its own selfish interest. An example is a communication network, in which each edge is a computer that possibly belongs to a different person. Different computers have different transmission speeds, so every edge in

the network has a numeric weight equal to the number of milliseconds it takes to transmit a message. Our goal is to send a message between two points in the network in the shortest time possible. If we know the transmission-time of each computer (the weight of each edge), then we can use a standard shortest-paths algorithm. If we do not know the transmission times, then we have to ask each computer to tell us its transmission-time. But, the computers may be selfish: a computer might tell us that its transmission time is very long, so that we will not bother it with our messages. A possible solution to this problem is to use a variant of the VCG mechanism, which gives the computers an incentive to reveal their true weights.

7 Linear programming formulation

There is a natural linear programming formulation for the shortest path problem, given below. It is very simple compared to most other uses of linear programs in discrete optimization, however it illustrates connections to other concepts.

Given a directed graph (V, A) with source node s , target node t , and cost w_{ij} for each edge (i, j) in A , consider the program with variables x_{ij}

$$\begin{aligned} & \text{minimize } \sum_{ij \in A} w_{ij} x_{ij} \quad \text{subject to } x \geq 0 \\ & \text{and for all } i, \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1, & \text{if } i = s; \\ -1, & \text{if } i = t; \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

The intuition behind this is that x_{ij} is an indicator variable for whether edge (i, j) is part of the shortest path: 1 when it is, and 0 if it is not. We wish to select the set of edges with minimal weight, subject to the constraint that this set forms a path from s to t (represented by the equality constraint: for all vertices except s and t the number of incoming and outgoing edges that are part of the path must be the same (i.e., that it should be a path from s to t)).

This LP has the special property that it is integral; more specifically, every basic optimal solution (when one exists) has all variables equal to 0 or 1, and the set of edges whose variables equal 1 form an s - t dipath. See Ahuja et al.^[6] for one proof, although the origin of this approach dates back to mid-20th century.

The dual for this linear program is

$$\begin{aligned} & \text{maximize } y_t - y_s \quad \text{subject to for all } ij, y_j - y_i \leq \\ & w_{ij} \end{aligned}$$

and feasible duals correspond to the concept of a consistent heuristic for the A* algorithm for shortest

paths. For any feasible dual y the reduced costs $w'_{ij} = w_{ij} - y_j + y_i$ are nonnegative and A* essentially runs Dijkstra's algorithm on these reduced costs.

8 General algebraic framework on semirings: the algebraic path problem

Many problems can be framed as a form of the shortest path for some suitably substituted notions of addition along a path and taking the minimum. The general approach to these is to consider the two operations to be those of a **semiring**. Semiring multiplication is done along the path, and the addition is between paths. This general framework is known as the **algebraic path problem**.^{[7][8][9]}

Most of the classic shortest-path algorithms (and new ones) can be formulated as solving linear systems over such algebraic structures.^[10]

More recently, an even more general framework for solving these (and much less obviously related problems) has been developed under the banner of **valuation algebras**.^[11]

9 Shortest path in stochastic time-dependent networks

In real-life situations, the transportation network is usually stochastic and time-dependent. In fact, a traveler traversing a link daily may experience different travel times on that link due not only to the fluctuations in travel demand (origin-destination matrix) but also due to such incidents as work zones, bad weather conditions, accidents and vehicle breakdowns. As a result, a stochastic time-dependent (STD) network is a more realistic representation of an actual road network compared with the deterministic one.^[12]

Despite considerable progress during the course of the past decade, it remains a controversial question how an optimal path should be defined and identified in stochastic road networks. In other words, there is no unique definition of an optimal path under uncertainty. One possible and common answer to this question is to find a path with the minimum expected travel time. The main advantage of using this approach is that efficient shortest path algorithms introduced for the deterministic networks can be readily employed to identify the path with the minimum expected travel time in a stochastic network. However, the resulting optimal path identified by this approach may not be reliable, because this approach fails to address travel time variability. To tackle this issue some researchers use distribution of travel time instead

of expected value of it so they find the probability distribution of total traveling time using different optimization methods such as **dynamic programming** and **Dijkstra's algorithm**.^[13] These methods use **stochastic optimization**, specifically stochastic dynamic programming to find the shortest path in networks with probabilistic arc length.^[14] It should be noted that the concept of travel time reliability is used interchangeably with travel time variability in the transportation research literature, so that, in general, one can say that the higher the variability in travel time, the lower the reliability would be, and vice versa.

In order to account for travel time reliability more accurately, two common alternative definitions for an optimal path under uncertainty have been suggested. Some have introduced the concept of the most reliable path, aiming to maximize the probability of arriving on time or earlier than a given travel time budget. Others, alternatively, have put forward the concept of an α -reliable path based on which they intended to minimize the travel time budget required to ensure a pre-specified on-time arrival probability.

10 See also

- **Pathfinding**
- **IEEE 802.1aq**
- **Flow network**
- **Shortest path tree**
- **Euclidean shortest path**
- **Min-plus matrix multiplication**
- **Bidirectional search**, an algorithm that finds the shortest path between two vertices on a directed graph

11 References

11.1 Notes

- [1] Sanders, Peter (March 23, 2009). "Fast route planning". Google Tech Talk.
- [2] Chen, Danny Z. (December 1996). "Developing algorithms and software for geometric path planning problems". *ACM Computing Surveys*. **28** (4es): 18. doi:10.1145/242224.242246.
- [3] Abraham, Ittai; Fiat, Amos; Goldberg, Andrew V.; Werneck, Renato F. "Highway Dimension, Shortest Paths, and Provably Efficient Algorithms". *ACM-SIAM Symposium on Discrete Algorithms*, pages 782-793, 2010.

- [4] Abraham, Ittai; Dellinger, Daniel; Goldberg, Andrew V.; Werneck, Renato F. research.microsoft.com/pubs/142356/HL-TR.pdf “A Hub-Based Labeling Algorithm for Shortest Paths on Road Networks”. Symposium on Experimental Algorithms, pages 230–241, 2011.
- [5] Kroger, Martin (2005). “Shortest multiple disconnected path for the analysis of entanglements in two- and three-dimensional polymeric systems”. *Computer Physics Communications*. **168** (168): 209–232. doi:10.1016/j.cpc.2005.01.020.
- [6] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin (1993). *Network Flows: Theory, Algorithms and Applications*. Prentice Hall. ISBN 0-13-617549-X.
- [7] John Baras; George Theodorakopoulos (4 April 2010). *Path Problems in Networks*. Morgan & Claypool Publishers. pp. 9–. ISBN 978-1-59829-924-3.
- [8] Mehryar Mohri, “Semiring frameworks and algorithms for shortest-distance problems”, *Journal of Automata, Languages and Combinatorics*, Volume 7 Issue 3, January 2002, Pages 321 - 350
- [9] <http://www.iam.unibe.ch/~{ }run/talks/2008-06-05-Bern-Jonczy.pdf>
- [10] Michel Gondran; Michel Minoux (2008). *Graphs, Dioids and Semirings: New Models and Algorithms*. Springer Science & Business Media. chapter 4. ISBN 978-0-387-75450-5.
- [11] Marc Pouly; Jürg Kohlas (2011). *Generic Inference: A Unifying Theory for Automated Reasoning*. John Wiley & Sons. Chapter 6. Valuation Algebras for Path Problems. ISBN 978-1-118-01086-0.
- [12] Rajabi-Bahaabadi, Mojtaba; Shariat-Mohaymany, Afshin; Babaei, Mohsen; Ahn, Chang Wook (2015). “Multi-objective path finding in stochastic time-dependent road networks using non-dominated sorting genetic algorithm”. *Expert Systems with Applications*. **42** (12): 5056–5064. doi:10.1016/j.eswa.2015.02.046.
- [13] Olya, Mohammad Hessam (2014). “Finding shortest path in a combined exponential – gamma probability distribution arc length”. *International Journal of Operational Research*. **21** (1): 25–37. doi:10.1504/IJOR.2014.064020.
- [14] Olya, Mohammad Hessam (2014). “Applying Dijkstra’s algorithm for general shortest path problem with normal probability distribution arc length”. *International Journal of Operational Research*. **21** (2): 143–154. doi:10.1504/IJOR.2014.064541.
- Bellman, Richard (1958). “On a routing problem”. *Quarterly of Applied Mathematics*. **16**: 87–90. MR 0102435.
- Cherkassky, Boris V.; Goldberg, Andrew V.; Radzik, Tomasz (1996). “Shortest paths algorithms: theory and experimental evaluation”. *Mathematical Programming*. Ser. A. **73** (2): 129–174. doi:10.1016/0025-5610(95)00021-6. MR 1392160.
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001) [1990]. “Single-Source Shortest Paths and All-Pairs Shortest Paths”. *Introduction to Algorithms* (2nd ed.). MIT Press and McGraw-Hill. pp. 580–642. ISBN 0-262-03293-7.
- Dantzig, G. B. (January 1960). “On the Shortest Route through a Network”. *Management Science*. **6** (2): 187–190. doi:10.1287/mnsc.6.2.187.
- Dijkstra, E. W. (1959). “A note on two problems in connexion with graphs” (PDF). *Numerische Mathematik*. **1**: 269–271. doi:10.1007/BF01386390.
- Ford, L. R. (1956). “Network Flow Theory”. Rand Corporation. P-923.
- Fredman, Michael Lawrence; Tarjan, Robert E. (1984). *Fibonacci heaps and their uses in improved network optimization algorithms*. 25th Annual Symposium on Foundations of Computer Science. IEEE. pp. 338–346. doi:10.1109/SFCS.1984.715934. ISBN 0-8186-0591-X.
- Fredman, Michael Lawrence; Tarjan, Robert E. (1987). “Fibonacci heaps and their uses in improved network optimization algorithms”. *Journal of the Association for Computing Machinery*. **34** (3): 596–615. doi:10.1145/28869.28874.
- Gabow, H. N. (1983). “Scaling algorithms for network problems”. *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS 1983)*. pp. 248–258. doi:10.1109/SFCS.1983.68.
- Gabow, Harold N. (1985). “Scaling algorithms for network problems”. *Journal of Computer and System Sciences*. **31** (2): 148–168. doi:10.1016/0022-0000(85)90039-X. MR 828519.
- Hagerup, Torben (2000). Montanari, Ugo; Rolim, José D. P.; Welzl, Emo, eds. *Improved Shortest Paths on the Word RAM*. *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*. pp. 61–72. ISBN 3-540-67715-1.
- Johnson, Donald B. (December 1981). “A priority queue in which initialization and queue operations

11.2 Bibliography

- Ahuja, Ravindra K.; Mehlhorn, Kurt; Orlin, James; Tarjan, Robert E. (April 1990). “Faster algorithms for the shortest path problem”. *Journal of the ACM*. ACM. **37** (2): 213–223. doi:10.1145/77600.77615.

- take $O(\log \log D)$ time”. *Mathematical Systems Theory*. **15** (1): 295–309. doi:10.1007/BF01786986. MR 683047.
- Karlsson, Rolf G.; Poblete, Patricio V. (1983). “An $O(m \log \log D)$ algorithm for shortest paths”. *Discrete Applied Mathematics*. **6** (1): 91–93. doi:10.1016/0166-218X(83)90104-X. MR 700028.
 - Leyzorek, M.; Gray, R. S.; Johnson, A. A.; Ladew, W. C.; Meaker, S. R., Jr.; Petry, R. M.; Seitz, R. N. (1957). *Investigation of Model Techniques — First Annual Report — 6 June 1956 — 1 July 1957 — A Study of Model Techniques for Communication Systems*. Cleveland, Ohio: Case Institute of Technology.
 - Moore, E. F. (1959). “The shortest path through a maze”. *Proceedings of an International Symposium on the Theory of Switching (Cambridge, Massachusetts, 2–5 April 1957)*. Cambridge: Harvard University Press. pp. 285–292.
 - Pettie, Seth; Ramachandran, Vijaya (2002). *Computing shortest paths with comparisons and additions*. *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*. pp. 267–276. ISBN 0-89871-513-X.
 - Pettie, Seth (26 January 2004). “A new approach to all-pairs shortest paths on real-weighted graphs”. *Theoretical Computer Science*. **312** (1): 47–74. doi:10.1016/s0304-3975(03)00402-x.
 - Pollack, M.; Wiebenson, W. (March–April 1960). “Solution of the Shortest-Route Problem—A Review”. *Op. Res.* **8** (2): 224–230. doi:10.1287/opre.8.2.224.
 - Schrijver, Alexander (2004). *Combinatorial Optimization — Polyhedra and Efficiency*. Algorithms and Combinatorics. **24**. Springer. ISBN 3-540-20456-3. Here: vol.A, sect.7.5b, p. 103
 - Shimbel, Alfonso (1953). “Structural parameters of communication networks”. *Bulletin of Mathematical Biophysics*. **15** (4): 501–507. doi:10.1007/BF02476438.
 - Thorup, Mikkel (1999). “Undirected single-source shortest paths with positive integer weights in linear time”. *Journal of the ACM (JACM)*. **46** (3): 362–394. doi:10.1145/316542.316548. Retrieved 28 November 2014.
 - Thorup, Mikkel (2004). “Integer priority queues with decrease key in constant time and the single source shortest paths problem”. *Journal of Computer and System Sciences*. **69** (3): 330–353. doi:10.1016/j.jcss.2004.04.003.
 - Whiting, P. D.; Hillier, J. A. (March–June 1960). “A Method for Finding the Shortest Route through a Road Network”. *Operational Research Quarterly*. **11** (1/2): 37–40. doi:10.1057/jors.1960.32.
 - Williams, Ryan (2014). “Faster all-pairs shortest paths via circuit complexity”. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC '14)*. New York: ACM. pp. 664–673. arXiv:1312.6680. doi:10.1145/2591796.2591811. MR 3238994.

11.3 Missing references

- Dantzig (1958). Missing or empty |title= (help)

12 Further reading

- Frigioni, D.; Marchetti-Spaccamela, A.; Nanni, U. (1998). “Fully dynamic output bounded single source shortest path problem”. *Proc. 7th Annu. ACM-SIAM Symp. Discrete Algorithms*. Atlanta, GA. pp. 212–221.
- Dreyfus, S. E. (October 1967). *An Appraisal of Some Shortest Path Algorithms* (PDF) (Report). Project Rand. United States Air Force. RM-5433-PR. DTIC AD-661265.

13 Text and image sources, contributors, and licenses

13.1 Text

- **Shortest path problem** *Source:* https://en.wikipedia.org/wiki/Shortest_path_problem?oldid=764952307 *Contributors:* AxelBoldt, General Wesc, LC~enwiki, The Anome, Shd~enwiki, JeLuF, B4hand, Michael Hardy, Chris-martin, Ijon, Dcoetzee, Dysprosia, Furrykef, Pigorsch, Altenmann, Lzur, Sho Uemura, Giftlite, Wolfkeeper, BenFrantzDale, Andris, Lqs, OverlordQ, Andreas Kaufmann, Rich Farmbrough, Rasmusdf, DcoetzeeBot~enwiki, ESkog, TerraFrost, Brian0918, MisterSheik, Caesura, Oleg Alexandrov, Nuno Tavares, Mindmatrix, Camw, Oliphaunt, Ruud Koot, BD2412, Qwertyus, Rjwilmsi, Xperimental~enwiki, Mathbot, Mathiasstck, Choess, Kri, Cthe, Chobot, Phelanpt, Chris Capoccia, Gaius Cornelius, Anomie, Nethgurb, Lt-wiki-bot, Cedar101, Treesmill, SmackBot, Meld, Ohnoitsjamie, DHN-bot~enwiki, Tommyjb, Lpgeffen, RomanSpa, MadScientistVX, Optakeover, Graph Theory page blanker, Devis, Headbomb, Magioladitis, Squire55, David Eppstein, Kope, Glrx, R'n'B, Edepot, Essess, Dmforcier, Alcidesfonseca, Aaron Rotenberg, Alfredo J. Herrera Lago, Kbrose, SieBot, Jan Winnicki, Taemyr, Lourakis, Jdaloner, Hariva, Denisarona, Justin W Smith, Metaprimer, Cairomax, Daveagp, Johnunig, Marc van Leeuwen, C. lorenz, Addbot, Fgnievinski, Jason.surratt, Download, Delaszk, Cipher1024, WikiDreamer Bot, Jarble, Luckas-bot, Yobot, AnomieBOT, Erel Segal, Materialsscientist, Citation bot, Xqbot, Alexander Anoprienko, Pmlineditor, Suzhouwuyue, Captain-n00dle, Deanphd, Citation bot 1, RedBot, Serols, Robert Geisberger, RobinK, Mjs1991, Trappist the monk, MoreNet, Horcrux92, ToneDaBass, John of Reading, WikitanvirBot, Super48paul, RA0808, Hari6389, Fæ, Mkroeger, Templatetypedef, ClueBot NG, MiroBrada, Nullzero, BG19bot, Marcelkcs, Happyuk, BattyBot, Jerry Hintze, Jochen Burghardt, Simonpratt, Amine.marref, HoboMcJoe, Ginsuloft, Robmccoll, Artyom Kalinin, Yacs, Monkbob, JMP EAX, Boky90, ToonLucas22, HelpUsStopSpam, KasparBot, CAPTAIN RAJU, Danieloliveira56, Mikkel2thorup, Bahaabadi, Bender the Bot and Anonymous: 130

13.2 Images

- **File:6n-graf.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/5/5b/6n-graf.svg> *License:* Public domain *Contributors:* Image:6n-graf.png similar input data *Original artist:* User:AzaToth
- **File:Lock-green.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/6/65/Lock-green.svg> *License:* CC0 *Contributors:* en:File:Free-to-read_lock_75.svg *Original artist:* User:Trappist the monk
- **File:Question_book-new.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/9/99/Question_book-new.svg *License:* Cc-by-sa-3.0 *Contributors:* Created from scratch in Adobe Illustrator. Based on Image:Question book.png created by User:Equazcion *Original artist:* Tkgd2007
- **File:Shortest_path_with_direct_weights.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3b/Shortest_path_with_direct_weights.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Artyom Kalinin
- **File:Text_document_with_red_question_mark.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a4/Text_document_with_red_question_mark.svg *License:* Public domain *Contributors:* Created by bdesham with Inkscape; based upon Text-x-generic.svg from the Tango project. *Original artist:* Benjamin D. Esham (bdesham)
- **File:Wiki_letter_w_cropped.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Wiki_letter_w_cropped.svg *License:* CC-BY-SA-3.0 *Contributors:* This file was derived from Wiki letter w.svg: `` *Original artist:* Derivative work by Thumperward

13.3 Content license

- Creative Commons Attribution-Share Alike 3.0