References

[1] E. H. L. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, 1989.

[2] J. Abello, A. Buchsbaum, and J. Westbrook. A functional approach to external graph algorithms. *Algorithmica*, 32(3):437–458, 2002.

[3] W. Ackermann. Zum hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99:118–133, 1928.

[4] G. M. Adel'son-Vel'skii and E. M. Landis. An algorithm for the organization of information. *Soviet Mathematics Doklady*, 3:1259–1263, 1962.

[5] A. Aggarwal and J. S. Vitter. The input/output complexity of sorting and related problems. *Communications of the ACM*, 31(9):1116–1127, 1988.

[6] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[7] A. V. Aho, B. W. Kernighan, and P. J. Weinberger. *The AWK Programming Language*. Addison-Wesley, 1988.

[8] R. K. Ahuja, R. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.

[9] R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan. Faster algorithms for the shortest path problem. *Journal of the ACM*, 3(2):213–223, 1990.

[10] N. Alon, M. Dietzfelbinger, P. B. Miltersen, E. Petrank, and E. Tardos. Linear hash functions. *Journal of the ACM*, 46(5):667–683, 1999.

[11] A. Andersson, T. Hagerup, S. Nilsson, and R. Raman. Sorting in linear time? *Journal of Computer and System Sciences*, 57(1):74–93, 1998.

[12] F. Annexstein, M. Baumslag, and A. Rosenberg. Group action graphs and parallel architectures. *SIAM Journal on Computing*, 19(3):544–569, 1990.

[13] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.

[14] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.

[15] H. Bast, S. Funke, P. Sanders, and D. Schultes. Fast routing in road networks with transit nodes. *Science*, 316(5824):566, 2007.

[16] R. Bayer and E. M. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1(3):173–189, 1972.

[17] R. Beier and B. Vöcking. Random knapsack in expected polynomial time. *Journal of Computer and System Sciences*, 69(3):306–329, 2004.

[18] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[19] M. A. Bender, E. D. Demaine, and M. Farach-Colton. Cache-oblivious B-trees. In *41st Annual Symposium on Foundations of Computer Science*, pages 399–409, 2000.

[20] J. L. Bentley and M. D. McIlroy. Engineering a sort function. *Software Practice and Experience*, 23(11):1249–1265, 1993.

[21] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, pages 643–647, 1979.

[22] J. L. Bentley and R. Sedgewick. Fast algorithms for sorting and searching strings. In *8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 360–369, 1997.

[23] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[24] G. E. Blelloch, C. E. Leiserson, B. M. Maggs, C. G. Plaxton, S. J. Smith, and M. Zagha. A comparison of sorting algorithms for the connection machine CM-2. In *3rd ACM Symposium on Parallel Algorithms and Architectures*, pages 3–16, 1991.

[25] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448, 1972.

[26] N. Blum and K. Mehlhorn. On the average number of rebalancing operations in weight-balanced trees. *Theoretical Computer Science*, 11:303–320, 1980.

[27] Boost.org. Boost C++ Libraries. `www.boost.org`.

[28] O. Boruvka. O jistém problému minimálním. *Pràce, Moravské Prirodovedecké Spolecnosti*, pages 1–58, 1926.

[29] F. C. Botelho, R. Pagh, and N. Ziviani. Simple and space-efficient minimal perfect hash functions. In *10th Workshop on Algorithms and Data Structures*, volume 4619 of Lecture Notes in Computer Science, pages 139–150. Springer, 2007.

[30] G. S. Brodal. Worst-case efficient priority queues. In *7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 52–58, 1996.

[31] G. S. Brodal and J. Katajainen. Worst-case efficient external-memory priority queues. In *6th Scandinavian Workshop on Algorithm Theory*, volume 1432 of Lecture Notes in Computer Science, pages 107–118. Springer, 1998.

[32] M. R. Brown and R. E. Tarjan. Design and analysis of a data structure for representing sorted lists. *SIAM Journal of Computing*, 9:594–614, 1980.

[33] R. Brown. Calendar queues: A fast $O(1)$ priority queue implementation for the simulation event set problem. *Communications of the ACM*, 31(10):1220–1227, 1988.

[34] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, Apr. 1979.

[35] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47:1028–1047, 2000.

[36] B. Chazelle and L. J. Guibas. Fractional cascading: I. A data structuring technique. *Algorithmica*, 1(2):133–162, 1986.

[37] B. Chazelle and L. J. Guibas. Fractional cascading: II. Applications. *Algorithmica*, 1(2):163–191, 1986.

[38] J.-C. Chen. Proportion extend sort. *SIAM Journal on Computing*, 31(1):323–330, 2001.

[39] J. Cheriyan and K. Mehlhorn. Algorithms for dense graphs and networks. *Algorithmica*, 15(6):521–549, 1996.

[40] B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest path algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73:129–174, 1996.

[41] E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 46–93. PWS, 1997.

[42] D. Cohen-Or, D. Levin, and O. Remez. Progressive compression of arbitrary triangular meshes. In *IEEE Conference on Visualization*, pages 67–72, 1999.

[43] S. A. Cook. *On the Minimum Computation Time of Functions*. PhD thesis, Harvard University, 1966.

[44] S. A. Cook. The complexity of theorem proving procedures. In *3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[45] G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. Wiley, 1951.

[46] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry – Algorithms and Applications*. Springer, 2nd edition, 2000.

[47] R. Dementiev, L. Kettner, J. Mehnert, and P. Sanders. Engineering a sorted list data structure for 32 bit keys. In *6th Workshop on Algorithm Engineering & Experiments*, New Orleans, 2004.

[48] R. Dementiev, L. Kettner, and P. Sanders. STXXL: Standard Template Library for XXL data sets. *Software: Practice and Experience*, 2007. To appear, see also `http://stxxl.sourceforge.net/`.

[49] R. Dementiev and P. Sanders. Asynchronous parallel disk sorting. In *15th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 138–148, San Diego, 2003.

[50] R. Dementiev, P. Sanders, D. Schultes, and J. Sibeyn. Engineering an external memory minimum spanning tree algorithm. In *IFIP TCS*, Toulouse, 2004.

[51] L. Devroye. A note on the height of binary search trees. *Journal of the ACM*, 33:289–498, 1986.

[52] R. B. Dial. Shortest-path forest with topological ordering. *Communications of the ACM*, 12(11):632–633, 1969.

[53] M. Dietzfelbinger, T. Hagerup, J. Katajainen, and M. Penttonen. A reliable randomized algorithm for the closest-pair problem. *Journal of Algorithms*, 1(25):19–51, 1997.

[54] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert, and R. E. Tarjan. Dynamic perfect hashing: Upper and lower bounds. *SIAM Journal of Computing*, 23(4):738–761, 1994.

[55] M. Dietzfelbinger and C. Weidling. Balanced allocation and dictionaries with tightly packed constant size bins. *Theoretical Computer Science*, 380(1–2):47–68, 2007.

[56] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[57] E. A. Dinic. Economical algorithms for finding shortest paths in a network. In *Transportation Modeling Systems*, pages 36–44, 1978.

[58] W. Domschke and A. Drexl. *Einführung in Operations Research*. Springer, 2007.

[59] J. R. Driscoll, N. Sarnak, D. D. Sleator, and R. E. Tarjan. Making data structures persistent. *Journal of Computer and System Sciences*, 38(1):86–124, 1989.

[60] J. Fakcharoenphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *Journal of Computer and System Sciences*, 72(5):868–889, 2006.

[61] R. Fleischer. A tight lower bound for the worst case of Bottom-Up-Heapsort. *Algorithmica*, 11(2):104–115, 1994.

[62] R. Floyd. Assigning meaning to programs. In J. Schwarz, editor, *Mathematical Aspects of Computer Science*, pages 19–32. AMS, 1967.

[63] L. R. Ford. Network flow theory. Technical Report P-923, Rand Corporation, Santa Monica, California, 1956.

[64] E. Fredkin. Trie memory. *Communications of the ACM*, 3:490–499, 1960.

[65] M. L. Fredman. On the efficiency of pairing heaps and related data structures. *Journal of the ACM*, 46(4):473–501, 1999.

[66] M. L. Fredman, J. Komlos, and E. Szemeredi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31:538–544, 1984.

[67] M. L. Fredman, R. Sedgewick, D. D. Sleator, and R. E. Tarjan. The pairing heap: A new form of self-adjusting heap. *Algorithmica*, 1:111–129, 1986.

[68] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34:596–615, 1987.

[69] M. Frigo, C. E. Leiserson, H. Prokop, and S. Ramachandran. Cache-oblivious algorithms. In *40th IEEE Symposium on Foundations of Computer Science*, pages 285–298, 1999.

[70] H. N. Gabow. Path-based depth-first search for strong and biconnected components. *Information Processing Letters*, pages 107–114, 2000.

[71] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[72] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[73] B. Gärtner and J. Matousek. *Understanding and Using Linear Programming*. Springer, 2006.

[74] GMP (GNU Multiple Precision Arithmetic Library). `http://gmplib.org/`.

[75] A. V. Goldberg. Scaling algorithms for the shortest path problem. *SIAM Journal on Computing*, 24:494–504, 1995.

[76] A. V. Goldberg. A simple shortest path algorithm with linear average time. In *9th European Symposium on Algorithms*, volume 2161 of Lecture Notes in Computer Science, pages 230–241. Springer, 2001.

[77] A. V. Goldberg and C. Harrelson. Computing the shortest path: A^* meets graph theory. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 156–165, 2005.

[78] M. T. Goodrich and R. Tamassia. JDSL – the data structures library in Java. `http://www.jdsl.org/`.

[79] G. Graefe and P.-A. Larson. B-tree indexes and CPU caches. In *17th International Conference on Data Engineering*, pages 349–358. IEEE, 2001.

[80] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.

[81] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 2nd edition, 1994.

[82] J. F. Grantham and C. Pomerance. Prime numbers. In K. H. Rosen, editor, *Handbook of Discrete and Combinatorial Mathematics*, chapter 4.4, pages 236–254. CRC Press, 2000.

[83] R. Grossi and G. Italiano. Efficient techniques for maintaining multidimensional keys in linked data structures. In *26th International Colloquium on Automata, Languages and Programming*, volume 1644 of Lecture Notes in Computer Science, pages 372–381. Springer, 1999.

[84] S. Halperin and U. Zwick. Optimal randomized EREW PRAM algorithms for finding spanning forests and for other basic graph connectivity problems. In *7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 438–447, 1996.

[85] Y. Han and M. Thorup. Integer sorting in $\mathrm{O}\left(n\sqrt{\log\log n}\right)$ expected time and linear space. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 135–144, 2002.

[86] G. Handler and I. Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10:293–309, 1980.

[87] J. Hartmanis and J. Simon. On the power of multiplication in random access machines. In *5th IEEE Symposium on Foundations of Computer Science*, pages 13–23, 1974.

[88] M. Held and R. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[89] M. Held and R. Karp. The traveling-salesman problem and minimum spanning trees, part II. *Mathematical Programming*, 1:6–25, 1971.

[90] P. V. Hentenryck and L. Michel. *Constraint-Based Local Search*. MIT Press, 2005.

[91] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:576–585, 1969.

[92] C. A. R. Hoare. Proof of correctness of data representations. *Acta Informatica*, 1:271–281, 1972.

[93] R. D. Hofstadter. Metamagical themas. *Scientific American*, pages 16–22, January 1983.

[94] P. Høyer. A general technique for implementation of efficient priority queues. In *3rd Israeli Symposium on Theory of Computing and Systems*, pages 57–66, 1995.

[95] S. Huddlestone and K. Mehlhorn. A new data structure for representing sorted lists. *Acta Informatica*, 17:157–184, 1982.

[96] J. Iacono. Improved upper bounds for pairing heaps. In *7th Scandinavian Workshop on Algorithm Theory*, volume 1851 of Lecture Notes in Computer Science, pages 32–45. Springer, 2000.

[97] A. Itai, A. G. Konheim, and M. Rodeh. A sparse table implementation of priority queues. In *8th International Colloquium on Automata, Languages and Programming*, volume 115 of Lecture Notes in Computer Science, pages 417–431. Springer, 1981.

[98] V. Jarník. O jistém problému minimálním. *Práca Moravské Přírodovědecké Společnosti*, 6:57–63, 1930.

[99] K. Jensen and N. Wirth. *Pascal User Manual and Report: ISO Pascal Standard*. Springer, 1991.

[100] T. Jiang, M. Li, and P. Vitányi. Average-case complexity of shellsort. In *26th International Colloquium on Automata, Languages and Programming*, volume 1644 of Lecture Notes in Computer Science, pages 453–462. Springer, 1999.

[101] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: Experimental evaluation; part II, graph coloring and number partitioning. *Operations Research*, 39(3):378–406, 1991.

[102] K. Kaligosi and P. Sanders. How branch mispredictions affect quicksort. In *14th European Symposium on Algorithms*, volume 4168 of Lecture Notes in Computer Science, pages 780–791. Springer, 2006.

[103] H. Kaplan and R. E. Tarjan. New heap data structures. Technical Report TR-597-99, Princeton University, 1999.

[104] A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics Doklady*, 7(7):595–596, 1963.

[105] D. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm for finding minimum spanning trees. *Journal of the ACM*, 42:321–329, 1995.

[106] N. Karmakar. A new polynomial-time algorithm for linear programming. *Combinatorica*, pages 373–395, 1984.

[107] J. Katajainen and B. B. Mortensen. Experiences with the design and implementation of space-efficient deque. In *Workshop on Algorithm Engineering*, volume 2141 of Lecture Notes in Computer Science, pages 39–50. Springer, 2001.

[108] I. Katriel, P. Sanders, and J. L. Träff. A practical minimum spanning tree algorithm using the cycle property. Technical Report MPI-I-2002-1-003, MPI Informatik, Germany, October 2002.

[109] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[110] L. Khachiyan. A polynomial time algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

[111] V. King. A simpler minimum spanning tree verification algorithm. *Algorithmica*, 18:263–270, 1997.

[112] D. E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, 2nd edition, 1998.

[113] D. E. Knuth. *MMIXware: A RISC Computer for the Third Millennium*, volume 1750 of Lecture Notes in Computer Science. Springer, 1999.

[114] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27:97–109, 1985.

[115] B. Korte and J.Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2000.

[116] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50, 1956.

[117] E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem*. Wiley, 1985.

[118] LEDA (Library of Efficient Data Types and Algorithms). `www.algorithmic-solutions.com`.

[119] L. Q. Lee, A. Lumsdaine, and J. G. Siek. *The Boost Graph Library: User Guide and Reference Manual*. Addison-Wesley, 2002.

[120] L. Levin. Universal search problems. *Problemy Peredachi Informatsii*, 9(3):265–266, 1973.

[121] I. Lustig and J.-F. Puget. Program does not equal program: Constraint programming and its relationship to mathematical programming. *Interfaces*, 31:29–53, 2001.

[122] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, 1990.

[123] C. Martínez and S. Roura. Optimal sampling strategies in Quicksort and Quickselect. *SIAM Journal on Computing*, 31(3):683–705, 2002.

[124] C. McGeoch, P. Sanders, R. Fleischer, P. R. Cohen, and D. Precup. Using finite experiments to study asymptotic performance. In *Experimental Algorithmics — From Algorithm Design to Robust and Efficient Software*, volume 2547 of Lecture Notes in Computer Science, pages 1–23. Springer, 2002.

[125] MCSTL: The Multi-Core Standard Template Library. `http://algo2.iti.uni-karlsruhe.de/singler/mcstl/`.

[126] K. Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27(3):125–128, Mar. 1988.

[127] K. Mehlhorn. Amortisierte Analyse. In T. Ottmann, editor, *Prinzipien des Algorithmenentwurfs*, pages 91–102. Spektrum Lehrbuch, 1998.

[128] K. Mehlhorn and U. Meyer. External memory breadth-first search with sublinear I/O. In *10th European Symposium on Algorithms*, volume 2461 of Lecture Notes in Computer Science, pages 723–735. Springer, 2002.

[129] K. Mehlhorn and S. Näher. Bounded ordered dictionaries in $O(\log\log N)$ time and $O(n)$ space. *Information Processing Letters*, 35(4):183–189, 1990.
[130] K. Mehlhorn and S. Näher. Dynamic fractional cascading. *Algorithmica*, 5:215–241, 1990.
[131] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.
[132] K. Mehlhorn, S. Näher, and P. Sanders. Engineering DFS-based graph algorithms. Submitted, 2007.
[133] K. Mehlhorn, V. Priebe, G. Schäfer, and N. Sivadasan. All-pairs shortest-paths computation in the presence of negative cycles. *Information Processing Letters*, 81(6):341–343, 2002.
[134] K. Mehlhorn and P. Sanders. Scanning multiple sequences via cache memory. *Algorithmica*, 35(1):75–93, 2003.
[135] K. Mehlhorn and M. Ziegelmann. Resource constrained shortest paths. In *8th European Symposium on Algorithms*, volume 1879 of Lecture Notes in Computer Science, pages 326–337, 2000.
[136] R. Mendelson, R. E. Tarjan, M. Thorup, and U. Zwick. Melding priority queues. In *9th Scandinavian Workshop on Algorithm Theory*, pages 223–235, 2004.
[137] *Meyers Konversationslexikon*. Bibliographisches Institut, 1888.
[138] B. Meyer. *Object-Oriented Software Construction*. Prentice Hall, 2nd edition, 1997.
[139] U. Meyer. Average-case complexity of single-source shortest-path algorithms: Lower and upper bounds. *Journal of Algorithms*, 48(1):91–134, 2003.
[140] U. Meyer and P. Sanders. Δ-stepping: A parallel shortest path algorithm. In *6th European Symposium on Algorithms*, number 1461 in Lecture Notes in Computer Science, pages 393–404. Springer, 1998.
[141] U. Meyer, P. Sanders, and J. Sibeyn, editors. *Algorithms for Memory Hierarchies*, volume 2625 of Lecture Notes in Computer Science. Springer, 2003.
[142] B. M. E. Moret and H. D. Shapiro. An empirical analysis of algorithms for constructing a minimum spanning tree. In *2nd Workshop on Algorithms and Data Structures*, volume 519 of Lecture Notes in Computer Science, pages 400–411. Springer, 1991.
[143] R. Morris. Scatter storage techniques. *Communications of the ACM*, 11(1):38–44, 1968.
[144] S. S. Muchnick. *Advanced Compiler Design and Implementation*. Morgan Kaufmann, 1997.
[145] S. Näher and O. Zlotowski. Design and implementation of efficient data types for static graphs. In *10th European Symposium on Algorithms*, volume 2461 of Lecture Notes in Computer Science, pages 748–759. Springer, 2002.
[146] G. Nemhauser and Z. Ullmann. Discrete dynamic programming and capital allocation. *Management Science*, 15(9):494–505, 1969.
[147] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.

[148] J. Nešetřil, H. Milková, and H. Nešetřilová. Otakar Boruvka on minimum spanning tree problem: Translation of both the 1926 papers, comments, history. *Discrete Mathematics*, 233(1–3):3–36, 2001.

[149] K. S. Neubert. The flashsort1 algorithm. *Dr. Dobb's Journal*, pages 123–125, February 1998.

[150] J. Nievergelt and E. Reingold. Binary search trees of bounded balance. *SIAM Journal of Computing*, 2:33–43, 1973.

[151] K. Noshita. A theorem on the expected complexity of Dijkstra's shortest path algorithm. *Journal of Algorithms*, 6(3):400–408, 1985.

[152] R. Pagh and F. Rodler. Cuckoo hashing. *Journal of Algorithms*, 51:122–144, 2004.

[153] W. W. Peterson. Addressing for random access storage. *IBM Journal of Research and Development*, 1(2), Apr. 1957.

[154] S. Pettie. Towards a final analysis of pairing heaps. In *46th IEEE Symposium on Foundations of Computer Science*, pages 174–183, 2005.

[155] S. Pettie and V. Ramachandran. An optimal minimum spanning tree algorithm. In *27th International Colloquium on Automata, Languages and Programming*, volume 1853 of Lecture Notes in Computer Science, pages 49–60. Springer, 2000.

[156] J. Pinkerton. *Voyages and Travels*, volume 2. 1808.

[157] P. J. Plauger, A. A. Stepanov, M. Lee, and D. R. Musser. *The C++ Standard Template Library*. Prentice Hall, 2000.

[158] R. C. Prim. Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, pages 1389–1401, Nov. 1957.

[159] W. Pugh. Skip lists: A probabilistic alternative to balanced trees. *Communications of the ACM*, 33(6):668–676, 1990.

[160] A. Ranade, S. Kothari, and R. Udupa. Register efficient mergesorting. In *High Performance Computing*, volume 1970 of Lecture Notes in Computer Science, pages 96–103. Springer, 2000.

[161] J. H. Reif. Depth-first search is inherently sequential. *Information Processing Letters*, 20(5):229–234, 1985.

[162] N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas. Efficiently four-coloring planar graphs. In *28th ACM Symposium on Theory of Computing*, pages 571–575, 1996.

[163] G. Robins and A. Zelikwosky. Improved Steiner tree approximation in graphs. In *11th ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.

[164] P. Sanders. Fast priority queues for cached memory. *ACM Journal of Experimental Algorithmics*, 5(7), 2000.

[165] P. Sanders and D. Schultes. Highway hierarchies hasten exact shortest path queries. In *13th European Symposium on Algorithms*, volume 3669 of Lecture Notes in Computer Science, pages 568–579. Springer, 2005.

[166] P. Sanders and D. Schultes. Engineering fast route planning algorithms. In *6th Workshop on Experimental Algorithms*, volume 4525 of Lecture Notes in Computer Science, pages 23–36. Springer, 2007.

[167] P. Sanders and S. Winkel. Super scalar sample sort. In *12th European Symposium on Algorithms*, volume 3221 of Lecture Notes in Computer Science, pages 784–796. Springer, 2004.

[168] R. Santos and F. Seidel. A better upper bound on the number of triangulations of a planar point set. *Journal of Combinatorial Theory, Series A*, 102(1):186–193, 2003.

[169] R. Schaffer and R. Sedgewick. The analysis of heapsort. *Journal of Algorithms*, 15:76–100, 1993.

[170] A. Schönhage. Storage modification machines. *SIAM Journal on Computing*, 9:490–508, 1980.

[171] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[172] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[173] D. Schultes. *Route Planning in Road Networks*. PhD thesis, 2008.

[174] R. Sedgewick. Analysis of shellsort and related algorithms. In *4th European Symposium on Algorithms*, volume 1136 of Lecture Notes in Computer Science, pages 1–11. Springer, 1996.

[175] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Addison-Wesley, 1996.

[176] R. Seidel and C. Aragon. Randomized search trees. *Algorithmica*, 16(4–5):464–497, 1996.

[177] R. Seidel and M. Sharir. Top-down analysis of path compression. *SIAM Journal of Computing*, 34(3):515–525, 2005.

[178] M. Sharir. A strong-connectivity algorithm and its applications in data flow analysis. *Computers and Mathematics with Applications*, 7(1):67–72, 1981.

[179] J. C. Shepherdson and H. E. Sturgis. Computability of recursive functions. *Journal of the ACM*, 10(2):217–255, 1963.

[180] J. Singler, P. Sanders, and F. Putze. MCSTL: The Multi-Core Standard Template Library. In *Euro-Par*, volume 4641 of Lecture Notes in Computer Science, pages 682–694. Springer, 2007.

[181] M. Sipser. *Introduction to the Theory of Computation*. MIT Press, 1998.

[182] D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983.

[183] D. D. Sleator and R. E. Tarjan. Self-adjusting binary search trees. *Journal of the ACM*, 32(3):652–686, 1985.

[184] D. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[185] R. E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160, 1972.

[186] R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

[187] R. E. Tarjan. Shortest paths. Technical report, AT&T Bell Laboratories, 1981.

[188] R. E. Tarjan. Amortized computational complexity. *SIAM Journal on Algebraic and Discrete Methods*, 6(2):306–318, 1985.

[189] R. E. Tarjan and U. Vishkin. An efficient parallel biconnectivity algorithm. *SIAM Journal on Computing*, 14(4):862–874, 1985.

[190] M. Thorup. Undirected single source shortest paths in linear time. *Journal of the ACM*, 46:362–394, 1999.

[191] M. Thorup. Even strongly universal hashing is pretty fast. In *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 496–497, 2000.

[192] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, 2004.

[193] M. Thorup. Integer priority queues with decrease key in constant time and the single source shortest paths problem. In *35th ACM Symposium on Theory of Computing*, pages 149–158, 2004.

[194] M. Thorup. 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, 2004.

[195] M. Thorup and U. Zwick. Approximate distance oracles. In *33rd ACM Symposium on the Theory of Computing*, pages 183–192, 2001.

[196] A. Toom. The complexity of a scheme of functional elements realizing the multiplication of integers. *Soviet Mathematics Doklady*, 150(3):496–498, 1963.

[197] Unknown. *Der Handlungsreisende – wie er sein soll und was er zu thun hat, um Auftraege zu erhalten und eines guecklichen Erfolgs in seinen Geschaeften gewiss zu sein – Von einem alten Commis-Voyageur*. 1832.

[198] P. van Emde Boas. Preserving order in a forest in less than logarithmic time. *Information Processing Letters*, 6(3):80–82, 1977.

[199] R. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, 2001.

[200] V. Vazirani. *Approximation Algorithms*. Springer, 2000.

[201] J. von Neumann. First draft of a report on the EDVAC. Technical report, University of Pennsylvania, 1945.

[202] J. Vuillemin. A data structure for manipulating priority queues. *Communications of the ACM*, 21:309–314, 1978.

[203] L. Wall, T. Christiansen, and J. Orwant. *Programming Perl*. O'Reilly, 3rd edition, 2000.

[204] I. Wegener. BOTTOM-UP-HEAPSORT, a new variant of HEAPSORT beating, on an average, QUICKSORT (if n is not very small). *Theoretical Computer Science*, 118(1):81–98, 1993.

[205] I. Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Springer, 2005.

[206] R. Wickremesinghe, L. Arge, J. S. Chase, and J. S. Vitter. Efficient sorting using registers and caches. *ACM Journal of Experimental Algorithmics*, 7(9), 2002.

[207] R. Wilhelm and D. Maurer. *Compiler Design*. Addison-Wesley, 1995.

[208] J. W. J. Williams. Algorithm 232: Heapsort. *Communications of the ACM*, 7:347–348, 1964.

Index