Algorithms Cheat Sheet

Time Complexity

Marcin Woch

Contents

Mathematical Operations

Arithmetics

Operation	Algorithm	Input	Output	Complexity
Addition	Schoolbook	Two n-digit numbers matrices	One $n+1$-digit number	$O(n)$
Subtraction	Schoolbook	Two n-digit numbers matrices	One $n+1$-digit number	$O(n)$
Multiplication	Schoolbook	Two n-digit numbers matrices	One $2n$-digit number	$O(n^2)$
	3-way Toom-Cook algorithm	Two n-digit numbers matrices	One $2n$-digit number	$O(n^{\log_3 5}) \approx O(n^{1.465})$
	k-way Toom-Cook algorithm	Two n-digit numbers matrices	One $2n$-digit number	$O\left(n^{\frac{\log(2k-1)}{\log k}}\right)$
	Mixed-level Toom-Cook algorithm	Two n-digit numbers matrices	One $2n$-digit number	$O(n2^{\sqrt{2\log n}}\log n)$
	Karatsuba algorithm	Two n-digit numbers matrices	One $2n$-digit number	$O(n^{\log_2 3}) \approx O(n^{1.585})$
	Schönhage–Strassen algorithm	Two n-digit numbers matrices	One $2n$-digit number	$O(n\log n\log\log n)$
	Harvey-Hoeven algorithm	Two n-digit numbers matrices	One $2n$-digit number	$O(n\log n)$
	Pointer machine [1]	Two n-digit numbers matrices	One $2n$-digit number	$O(n)$
	Unit Cost RAM machine [1]	Two n-digit numbers matrices	One $2n$-digit number	$O(n)$
Division	Schoolbook	Two n-digit numbers matrices	One n-digit number	$O(n^2)$
	Burnikel–Ziegler Divide-and-Conquer[2]	Two n-digit numbers matrices	One n-digit number	$O(M(n)\log n)$
	Newton–Raphson division[2]	Two n-digit numbers matrices	One n-digit number	$O(M(n))$
Square root	Newton's method[2]	One n-digit number	One n-digit number	$O(M(n))$
Modular exponentation	Repeated multiplication and reduction[2]	Two n-digit integers, k-bit exponent	One n-digit integer	$O(M(n)2^k)$
	Exponentiation by squaring[2]	Two n-digit integers, k-bit exponent	One n-digit integer	$O(M(n)k)$
	Exponentiation with Montgomery reduction[2]	Two n-digit integers, k-bit exponent	One n-digit integer	$O(M(n)k)$

[1] Theoretical model only

[2] $M(n)$ - The complexity of an implemented multiplication algorithm

Matrix Algebra

Operation	Algorithm	Input	Output	Complexity
Multiplication	Schoolbook	Two $n \times n$ matrices	One $n \times n$ matrix	$O(n^3)$
	Strassen's	Two $n \times n$ matrices	One $n \times n$ matrix	$O(n^{\log_2 7}) = O(n^{2.807})$
	Coppersmith-Winograd	Two $n \times n$ matrices	One $n \times n$ matrix	$O(n^{2.376})$
	Alman-Williams	Two $n \times n$ matrices	One $n \times n$ matrix	$O(n^{2.3728596})$
	Duan, Wu, Zhou	Two $n \times n$ matrices	One $n \times n$ matrix	$O(n^{2.3719})$
	Williams, Xu, Xu, Zhou	Two $n \times n$ matrices	One $n \times n$ matrix	$O(n^{2.3716})$
	Schoolbook	One $n \times m$ matrix, one $m \times p$ matrix	One $n \times p$ matrix	$O(nmp)$
Inversion	Gauss-Jordan elimination	One $n \times n$ matrix	One $n \times n$ matrix	$O(n^3)$
	Strassen algorithm	One $n \times n$ matrix	One $n \times n$ matrix	$O(n^{2.807})$
	Coppersmith-Winograd algorithm	One $n \times n$ matrix	One $n \times n$ matrix	$O(n^{2.376})$
	Optimised CW algorithm	One $n \times n$ matrix	One $n \times n$ matrix	$O(n^{2.373})$
SVD	Bidiagonalization, QR algorithm	One $m \times n$ matrix ($m \leq n$)	One $m \times m$ One $m \times n$ matrix One $n \times n$	$O(m^2n)$
	Laplace expansion	One $n \times n$ matrix	One number	$O(n!)$
	Division free algorithm	One $n \times n$ matrix	One number	$O(n^4)$
	LU decomposition	One $n \times n$ matrix	One number	$O(n^3)$
	Bareiss algorithm	One $n \times n$ matrix	One number	$O(n^3)$
	Fast matrix multiplication	One $n \times n$ matrix	One number	$O(n^{2.373})$
Back substitution	Back substitution algorithm	Triangular matrix	n solutions	$O(n^2)$

Polynomials

Operation	**Algorithm**	**Input**	**Output**	**Complexity**
Polynomial evaluation	Direct	One polynomial of degree n and integer coefficients	One number	$O(n)$
	Horner's algorithm	One polynomial of degree n and integer coefficients	One number	$O(n)$
Polynomial gcd	Euclid's algorithm	Two polynomials of degree n and integer coefficients	One number	$O(n^2)$
	Lehmer's algorithm (Fast Euclidean) [3]	Two polynomials of degree n and integer coefficients	One number	$O(M(n) \log n)$

[3] $M(n)$ - The complexity of an implemented multiplication algorithm

Number theory

Operation	Algorithm	Input	Output	Complexity
Greatest common divisor	Euclidean algorithm	Two n-digit integers	One integer	$O(n^2)$
	Binary GCD	Two n-digit integers	One integer	$O(n^2)$
	Left/right k-ary binary GCD	Two n-digit integers	One integer	$O(\frac{n^2}{\log n})$
	Stehle-Zimmermann algorithm[4]	Two n-digit integers	One integer	$O(M(n)\log n)$
	Schönhage algorithm[4]	Two n-digit integers	One integer	$O(M(n)\log n)$
Jacobi symbol	Stehle-Zimmermann algorithm[4]	Two n-digit integers	0, -1 or 1	$O(M(n)\log n)$
	Schönhage algorithm[4]	Two n-digit integers	0, -1 or 1	$O(M(n)\log n)$
Factorial	Bottom-up multiplication[4]	One positive integer less than n	One integer	$O(M(n^2)\log n)$
	Binary splitting[4]	One positive integer less than n	One integer	$O(M(n\log n)\log n)$
	Exponentiation of the prime factors of n[4]	One positive integer less than n	One integer	$O(M(n\log n)\log\log n)$
	Exponentiation of the prime factors of n[4]	One positive integer less than n	One integer	$O(M(n\log n))$
Primality test	AKS primality test n	n digit integer	True or false	$O(n^{6+O(1)})$
	AKS primality test with Agrawal's conjecture$n$	n digit integer	True or false	$O(n^3)$
	Elliptic curve test[5] - heuristical approach	n digit integer	True or false	$O(n^{4+\epsilon})$
	Baillie-PSW test[5]	n digit integer	True or false	$O(n^{2+\epsilon})$
	Miller-Rabin test[5] [6]	n digit integer	True or false	$O(kn^{2+\epsilon})$
	Solovay-Strassen test[5] [6]	n digit integer	True or false	$O(kn^{2+\epsilon})$
Integer factorisation	General number field sieve[5]	b-bit input integer	A set of factors	$O((1+\epsilon)^b)$
	Shor's algorithm [4] [7]	b-bit input integer	A set of factors	$O(M(b)b)$

[4] $M(n)$ - The complexity of an implemented multiplication algorithm
[5] ϵ - a positive constant
[6] k - a positive constant
[7] Theoretical model, on quantum computer

Additional Operations

Operation	Algorithm	Input	Output	Complexity
Discrete Fourier transform	Schoolbook	Size n data sequence	Set of complex number	$O(n^2)$
	Fast Fourier transform	Size n data sequence	Set of complex number	$O(n \log n)$
Golden ration	Newton's method[8]			$O(M(n))$
Square root of 2	Newton's method[8]			$O(M(n))$
Euler's number	Taylor series binary splitting of the exp. function[8]			$O(M(n) \log n)$
	Newton inversion of the natural logarithm[8]			$O(M(n) \log n)$
Pi	Arctan series binary splitting in Machin's formula[8]			$O(M(n) \log^2 n)$
	Gauss-Legendre algorithm[8]			$O(M(n) \log n)$
Euler's constant	Sweeney's method[8]			$O(M(n) \log^2 n)$
Gamma function	Approx. of the incomplete gamma function[8]	n digit number		$O(M(n)n^{\frac{1}{2}} \log^2 n)$
	Hypergeometric series [8]	Fixed ration number		$O(M(n) \log^2 n)$
Hypergeometric function	Borwein and Borwein [8]	n-digit number		$O(M(n)n^{\frac{1}{2}} \log^2 n)$
	Hypergeometric series [8]	Fixed rational number		$O(M(n) \log^2 n)$
Taylor series	Repeated argument reduction[8 9]			$O(M(n)n^{\frac{1}{2}})$
	FFT-based acceleration [8 9]			$O(M(n)n^{\frac{1}{3}} \log^2 n)$
	Binary splitting + bit-burst[8 9]			$O(M(n) \log^2 n)$
Arithmetic-geometric mean iteration	Arithmetic-geometric mean iteration[8 9]			$O(M(n) \log n)$

[8] $M(n)$ - The complexity of an implemented multiplication algorithm
[9] Aplicability: exp. log, sin, cos, arctan

Common Operations

Basic Data Structures

Data Structure	Operation	Average Time Complexity	Worst Time Complexity
Array	Access	$O(1)$	$O(1)$
	Deletion	$O(n)$	$O(n)$
	Insertion	$O(n)$	$O(n)$
	Search	$O(n)$	$O(n)$
Doubly-Linked List	Access	$O(n)$	$O(n)$
	Deletion	$O(1)$	$O(1)$
	Insertion	$O(1)$	$O(1)$
	Search	$O(n)$	$O(n)$
Hash Table	Deletion	$O(1)$	$O(n)$
	Insertion	$O(1)$	$O(n)$
	Search	$O(1)$	$O(n)$
Queue	Access	$O(n)$	$O(n)$
	Deletion	$O(1)$	$O(1)$
	Insertion	$O(1)$	$O(1)$
	Search	$O(n)$	$O(n)$
Singly-Linked List	Access	$O(n)$	$O(n)$
	Deletion	$O(1)$	$O(1)$
	Insertion	$O(1)$	$O(1)$
	Search	$O(n)$	$O(n)$
Skip List	Access	$O(\log n)$	$O(n)$
	Deletion	$O(\log n)$	$O(n)$
	Insertion	$O(\log n)$	$O(n)$
	Search	$O(\log n)$	$O(n)$
Stack	Access	$O(n)$	$O(n)$
	Deletion	$O(1)$	$O(1)$
	Insertion	$O(1)$	$O(1)$
	Search	$O(n)$	$O(n)$

Trees Data Structures

Data Structure	Operation	Average Time Complexity	Worst Time Complexity
AVL Tree	Access	$O(\log n)$	$O(\log n)$
	Deletion	$O(\log n)$	$O(\log n)$
	Insertion	$O(\log n)$	$O(\log n)$
	Search	$O(\log n)$	$O(\log n)$
Binary Search Tree	Access	$O(\log n)$	$O(n)$
	Deletion	$O(\log n)$	$O(n)$
	Insertion	$O(\log n)$	$O(n)$
	Search	$O(\log n)$	$O(n)$
B-Tree	Access	$O(\log n)$	$O(\log n)$
	Deletion	$O(\log n)$	$O(\log n)$
	Insertion	$O(\log n)$	$O(\log n)$
	Search	$O(\log n)$	$O(\log n)$
Cartesian Tree	Deletion	$O(\log n)$	$O(n)$
	Insertion	$O(\log n)$	$O(n)$
	Search	$O(\log n)$	$O(n)$
KD Tree	Access	$O(\log n)$	$O(n)$
	Deletion	$O(\log n)$	$O(n)$
	Insertion	$O(\log n)$	$O(n)$
	Search	$O(\log n)$	$O(n)$
Red-Black Tree	Access	$O(\log n)$	$O(\log n)$
	Deletion	$O(\log n)$	$O(\log n)$
	Insertion	$O(\log n)$	$O(\log n)$
	Search	$O(\log n)$	$O(\log n)$
Splay Tree	Deletion	$O(\log n)$	$O(\log n)$
	Insertion	$O(\log n)$	$O(\log n)$
	Search	$O(\log n)$	$O(\log n)$

Heap

Data Structure	Operation	Complexity
Binary Heap	Find Max	$O(1)$
	Extract Max	$O(\log n)$
	Increase Key	$O(\log n)$
	Insert	$O(\log n)$
	Delete	$O(\log n)$
	Merge	$O(n+m)$
Binomial Heap	Find Max	$O(1)$
	Extract Max	$O(\log n)$
	Increase Key	$O(\log n)$
	Insert	$O(1)$
	Delete	$O(\log n)$
	Merge	$O(\log n)$
Fibonacci Heap	Find Max	$O(1)$
	Extract Max	$O(\log n)$
	Increase Key	$O(1)$
	Insert	$O(1)$
	Delete	$O(\log n)$
	Merge	$O(1)$
Pairing Heap	Find Max	$O(1)$
	Extract Max	$O(\log n)$
	Increase Key	$O(\log n)$
	Insert	$O(1)$
	Delete	$O(\log n)$
	Merge	$O(1)$

Sorting Algorithms

Comparison Sorting Algorithms

Name	**Best**	**Average**	**Worst**
Block sort	$O(n)$	$O(n \log n)$	$O(n \log n)$
Bubble sort	$O(n)$	$O(n^2)$	$O(n^2)$
Cocktail shaker sort	$O(n)$	$O(n^2)$	$O(n^2)$
Comb sort[10]	$O(n \log n)$	$O(n^2/2^p)$	$O(n^2)$
Cubesort	$O(n)$	$O(n \log n)$	$O(n \log n)$
Cycle sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Exchange sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Gnome sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
In-place merge sort	–	-	$O(n \log^2 n)$
Insertion sort	$O(n)$	$O(n^2)$	$O(n^2)$
Introsort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Heapsort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Library sort	$O(n \log n)$	$O(n \log n)$	n^2
Merge sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Odd-even sort	$O(n)$	$O(n^2)$	$O(n^2)$
Patience sort	$O(n)$	$O(n \log n)$	$O(n \log n)$
Quicksort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Shellsort	$O(n \log n)$	$O(n \log^2 n)$	$O(n \log^2 n)$
Simple pancake sort	$O(n)$	$O(n^2)$	$O(n^2)$
Smoothsort	$O(n)$	$O(n \log n)$	$O(n \log n)$
Strand sort	$O(n)$	$O(n^2)$	$O(n^2)$
Timsort	$O(n)$	$O(n \log n)$	$O(n \log n)$
Tournament sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Tree (balanced) sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Tree (unbalanced) sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$

[10] p - number of increments

Non-comparison Sorting Algorithms

Name	Best	Average	Worst	Comment
Bucket sort (integer keys)[12]		$O(n)$	$O(n+r)$	If r is $O(n)$. Integers only. Uniform distribution of elements.
Bucket sort (integer keys)[12]		$O(n+r)$	$O(n+r)$	Integers only. Uniform distribution of elements.
Bucket sort (uniform keys)[13]		$O(n+k)$	$O(n^2 \times k)$	Integers only. Uniform distribution of elements.
Burstsort[13,14]		$O(\frac{nk}{d})$	$O(\frac{nk}{d})$	Can sort non-integers.
Counting sort[12]		$O(n+r)$	$O(n+r)$	Integers only.
Counting sort[12]		$O(n)$	$O(n+r)$	If r is $O(n)$. Integers only.
Flashsort[12]	$O(n)$	$O(n+r)$	$O(n^2)$	Best time achieved for uniform distribution of elements. For skewed distributions it can be quadratic.
In-place MSD Radix sort[13,14]		$O(nk)$	$O(nk)$	Can sort non-integers.
LSD Radix sort[13,14]	$O(n)$	$O(\frac{nk}{d})$	$O(\frac{nk}{d})$	Can sort non-integers.
MSD Radix sort[13,14]		$O(\frac{nk}{d})$	$O(\frac{nk}{d})$	Can sort non-integers.
Pigeonhole sort[13]		$O(n+2^k)$	$O(n+2^k)$	Integers only.
Postman sort[13,14]		$O(\frac{nk}{d})$	$O(\frac{nk}{d})$	
Spreadsort[13,14]	$O(n)$	$O(\frac{nk}{d})$	$O(n(\frac{k}{d}+d))$	Can sort non-integers.

[12] r - range of numbers to be sorted
[13] k - key size
[14] d - digit size

Other Sorting Algorithms

Below algorithms are presented for educational purposes only. The use of them is impractical in real-life situations due to very poor performance, like unbounded time in bogosort. For this reason only worst time complexity is presented here.

Name	Worst	Comment
Bead sort[15]	$O(S)$	Positive integers only, requires specialised hardware to achieve linear complexity.
Bogosort	Unbounded	
"I can't believe it can sort"	$O(n^2)$	
Merge-insertion sort	$O(n \log n)$	Implementation very complex.
Spaghetti sort/Poll sort	$O(n)$	Requires n parallel processors.
Stooge sort	$O(n^{\log_{1.5} 3})$	

[15] S - sum of all integers

Searching

Searching is a fundamental operation on data. Searching algorithms have already been described in Common Operations chapter where various data types imply different searcing time complexity. Below table lists only algorithms not mentioned earlier.

Name	**Worst**	**Comment**
Binary search	$O(\log n)$	Requires sorted data as an input
Linear search	$O(n)$	
Hashing	$O(n)$	Average constant time complexity

Graphs

Data Structure Opera0TBX12Aqz6eCu7gtions

Data Structure	**Operation**	**Complexity**
Adjacency list	Storage	$O(\|V\| + \|E\|)$
	Add vertex	$O(1)$
	Add edge	$O(1)$
	Remove vertex	$O(\|V\| + \|E\|)$
	Remove edge	$O(\|E\|)$
	Query	$O(\|V\|)$
Adjacency matrix	Storage	$O(\|V\|^2)$
	Add vertex	$O(\|V\|^2)$
	Add edge	$O(1)$
	Remove vertex	$O(\|V\|^2)$
	Remove edge	$O(1)$
	Query	$O(1)$
Incidence list	Storage	$O(\|V\| + \|E\|)$
	Add vertex	$O(1)$
	Add edge	$O(1)$
	Remove vertex	$O(\|E\|)$
	Remove edge	$O(\|E\|)$
	Query	$O(\|E\|)$
Incidence matrix	Storage	$O(\|V\| \times \|E\|)$
	Add vertex	$O(\|V\| \times \|E\|)$
	Add edge	$O(\|V\| \times \|E\|)$
	Remove vertex	$O(\|V\| \times \|E\|)$
	Remove edge	$O(\|V\| \times \|E\|)$
	Query	$O(\|E\|)$

Graph Search Algorithms

Operation	**Algorithm**	**Input**	**Comment**	**Complexity**
A* Graph Search	A*	b - branching factor, d - depth		$O(b^d)$
Explicit Graph Search	Depth First Search	V - vertices, E - edges		$O(\lvert V\rvert + \lvert E\rvert)$
	Breadth First Search	V - vertices, E - edges		$O(\lvert V\rvert + \lvert E\rvert)$
Implicit Graph Search	Depth First Search	b - branching factor, d - depth		$O(b^d)$
	Breadth First Search	b - branching factor, d - depth		$O(b^d)$

Other Graph Algorithms

Operation	**Algorithm**	**Input**	Data Structure	**Average Complexity**	**Worst Complexity**
Minimum Spanning Tree	Prim's Algorithm	V - vertices, E - edges		$O(\|E\| \log \|V\|)$	$O(\|V\|^2)$
Shortest Path	Bellman-Ford Algorithm	V - vertices, E - edges		$O(\|E\|\|V\|)$	$O(\|E\|\|V\|)$
	Dijkstra's Algorithm	V - vertices, E - edges	Priority queue/heap		$O(\|V\| + \|E\|) \log \|V\|$
	Dijkstra's Algorithm	V - vertices, E - edges	Array		$O(\|V\|^2)$
	Floyd-Warshall Algorithm	V - vertices, E - edges		$O(\|V\|^3)$	$O(\|V\|^3)$
Topological sort	Depth First Search	V - vertices, E - edges		$O(\|E\| + \|V\|)$	$O(\|E\| + \|V\|)$
	Kahn's Algorithm	V - vertices, E - edges		$O(\|E\| + \|V\|)$	$O(\|E\| + \|V\|)$

References

[1] Bradford P. G., Rawlins G. J. E., Shannon G. E. - Efficient matrix chain ordering in polylog time. SIAM Journal on Computing, 1998. 27 (2): 466–490. MR 1616556.

[2] Brent R. P., Zimmermann P. - Modern Computer Arithmetic. Cambridge University Press, 2010, ISBN 978-0-521-19469-3.

[3] Cormen T. H., Leiserson C. E. - Introduction to Algorithms, fourth edition, MIT Press, 2022.

[4] Chudnovsky D., Chudnovsky G. - Approximations and complex multiplication according to Ramanujan. Ramanujan revisited: Proceedings of the Centenary Conference. Academic Press, 1988 pp. 375–472. ISBN 978-0-01-205856-5.

[5] Knuth D. E. - Seminumerical Algorithms. The Art of Computer Programming Vol. 2 (3rd ed.), Addison-Wesley, 1997, ISBN 978-0-201-89684-8.

[6] Kuperberg G. - A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. SIAM Journal on Computing, 2005. Philadelphia. 35 (1): 188. arXiv:quant-ph/0302112. doi:10.1137.

[7] Miltersen P.B. - Derandomizing Complexity Classes. Handbook of Randomized Computing. Combinatorial Optimization. Kluwer Academic Pub., 2001, 9: 843. ISBN 978-1-4613-4886-3.

[8] Schönhage A., Grotefeld A.F.W., Vetter E. - Fast Algorithms—A Multitape Turing Machine Implementation. BI Wissenschafts-Verlag, 1994. ISBN 978-3-411-16891-0.

[9] Sedgewick R. - Algorithms, Addison Wesley Publishing Company, 1988.

[10] Sedgewick R., Wayne K. - Algorithms, Princeton University, 2011.

[11] Vassilevska W. V. - Breaking the Coppersmith-Winograd barrier: Multiplying matrices in $O(n^{2.373})$ time, 2014.