

Łagodne wprowadzenie do analizy algorytmów

Marek Kubale

Gdańsk 2025

PRZEWODNICZĄCY KOMITETU REDAKCYJNEGO
WYDAWNICTWA POLITECHNIKI GDAŃSKIEJ

Dariusz Mikielewicz

REDAKTOR

Zdzisław Pułaczewski

RECENZENT

Krzysztof Goczyła

Wydanie XX – 2024

Wydano za zgodą
Rektora Politechniki Gdańskiej

Oferta wydawnicza Politechniki Gdańskiej jest dostępna pod adresem
<https://sklep.pg.edu.pl>

Utwór nie może być powielany i rozpowszechniany, w jakiegokolwiek
formie i w jakiegokolwiek sposób, bez pisemnej zgody wydawcy.

© Copyright by Wydawnictwo Politechniki Gdańskiej, Gdańsk 2025

ISBN 978-83-7348-933-2

Spis treści

PRZEDMOWA	5
1. WPROWADZENIE.....	7
1.1. Rys historyczny	7
1.2. Klasyfikacja problemów.....	9
1.3. Język PseudoPascal.....	15
1.4. Podstawy matematyczne	17
1.4.1. Logarytmy i zaokrąglenia całkowite.....	17
1.4.2. Sumy szeregów	18
1.5. Symbole oszacowań asymptotycznych	20
1.5.1. Symbol $O(\cdot)$	21
1.5.2. Symbol $o(\cdot)$	21
1.5.3. Symbol $\Omega(\cdot)$	22
1.5.4. Symbol $\omega(\cdot)$	22
1.5.5. Symbol $\Theta(\cdot)$	23
1.5.6. Symbol $\tilde{\Theta}(\cdot)$	24
1.6. Równania rekurencyjne niejednorodne.....	24
1.6.1. Równania typu „dziel i zwyciężaj”	25
1.6.2. Równania typu „jeden krok w tył”	27
Zadania	31
2. PODSTAWY ANALIZY ALGORYTMÓW	38
2.1. Wstęp.....	38
2.2. Poprawność algorytmów	40
2.3. Złożoność czasowa algorytmów	43
2.3.1. Operacje podstawowe	43
2.3.2. Rozmiar danych	44
2.3.3. Pesymistyczna złożoność obliczeniowa	45
2.3.4. Oczekiwana złożoność obliczeniowa	45
2.4. Złożoność pamięciowa.....	48
2.5. Optymalność	52
2.6. Dokładność numeryczna algorytmów	54
2.6.1. Zadania źle uwarunkowane	54
2.6.2. Stabilność numeryczna	56
2.7. Prostota algorytmów	57
2.8. Wrażliwość numeryczna algorytmów	60
2.9. Programowanie a złożoność obliczeniowa	61
2.9.1. Rząd złożoności obliczeniowej.....	61
2.9.2. Stała proporcjonalności złożoności obliczeniowej	64
2.9.3. Imperatyw złożoności obliczeniowej i odstępstwa.....	67

2.10. Przykład analizy: mnożenie macierzy.....	68
2.11. Algorytmy probabilistyczne.....	72
Zadania	75
3. PODSTAWOWE STRUKTURY DANYCH	84
3.1. Tablice.....	84
3.2. Listy.....	86
3.3. Zbiory.....	87
3.4. Grafy	88
3.4.1. Macierz sąsiedztwa wierzchołków	94
3.4.2. Listy sąsiedztwa wierzchołków	97
3.4.3. Pęki wyjściowe	98
Zadania.....	98
4. DODATEK: ZAGADKI ALGORYTMICZNE.....	105
4.1. Alicja i Bogdan w pizzerii.....	105
4.2. Alicja i Bogdan w naleśnikarni	106
4.3. Alicja i Bogdan w samochodzie.....	107
4.4. Alicja i Bogdan wśród ludożerców	107
4.5. Alicja i Bogdan remontują mieszkanie	108
4.6. Alicja i Bogdan na działce	109
4.7. Alicja i Bogdan zostają deweloperami.....	109
4.8. Alicja kupuje buty	110
4.9. Alicja i Bogdan wyprawiają wesele.....	110
4.10. Alicja w krainie czarów	111
4.11. Alicja i Bogdan w kapeluszach	111
4.12. Alicja i Bogdan na wczasach	112
4.13. Kilka praktycznych problemów Alicji i Bogdana.....	112
4.14. Bogdan w więzieniu	112
SŁOWNIK POLSKO-ANGIELSKI.....	114
LITERATURA.....	119

PRZEDMOWA

Przekazywana do rąk czytelników książka jest 21. wydaniem podręcznika akademickiego, opublikowanego nakładem Wydawnictwa Politechniki Gdańskiej pod tym samym tytułem. Od momentu poprzedniego wydania wystąpiły nowe fakty w zakresie komputerowego stymulowania odkryć w dziedzinie teorii algorytmów, szybkiego mnożenia macierzy oraz rosnącej roli stabilności numerycznej algorytmów. Stąd zrodziła się potrzeba kolejnego wydania, uzupełnionego o najnowsze informacje z tych dziedzin oraz nowe zadania algorytmiczne.

Oddawany do rąk czytelników podręcznik jest przeznaczony dla osób interesujących się podstawami informatyki, w tym przede wszystkim dla studentów kierunku informatyka na Wydziale ETI Politechniki Gdańskiej. Formalnie rzecz biorąc, jego treść pokrywa pierwszą część wykładu z przedmiotu „Podstawy analizy algorytmów”, tj. algorytmy i problemy wielomianowe i całość wykładu z przedmiotu „Elementy analizy algorytmów”, ale stanowi też miejscami rozszerzenie programu tych przedmiotów, które są prowadzone na II roku kierunku informatyka. W tym miejscu odnotujmy, że drugą część wykładu doskonale pokrywa książka Krzysztofa Giary „Złożoność obliczeniowa algorytmów w zadaniach” [10] oraz dawny skrypt autora [13]. W szczególności niniejszy podręcznik może służyć jako wprowadzenie do wykładu „Algorytmy i struktury danych”. Jego fragmenty mogą być także wykorzystane w nauczaniu przedmiotu „Matematyka dyskretna”. Sądzę, że książka może ponadto zainteresować studentów kierunku informatyka na Wydziale Matematyki, Fizyki i Informatyki Uniwersytetu Gdańskiego, oraz studentów kierunków pokrewnych, np. matematyka stosowana.

Zakładam, że czytelnik ma pewne podstawowe przygotowanie z matematyki dyskretnej i że umie układać algorytmy w Pascalu lub dowolnym innym języku wysokiego poziomu. Znajomość przedmiotów „Metody i techniki programowania”, „Praktyka programowania” oraz „Matematyka dyskretna” jest pożądana, aczkolwiek niekonieczna, przy lekturze tego podręcznika akademickiego.

Niniejsza pozycja składa się z trzech rozdziałów zasadniczych. Rozdział 1 daje podstawy formalne, niezbędne przy analizie algorytmów pod kątem złożoności obliczeniowej. Podajemy tutaj klasyfikację problemów rozwiązywalnych za pomocą komputerów, przypominamy wybrane pojęcia matematyczne, definiujemy symbole oszacowań asymptotycznych. Jednakże najwięcej miejsca poświęcamy metodom najczęściej spotykanym przy analizie złożoności obliczeniowej algorytmów rekurencyjnych.

Rozdział 2 wprowadza w zagadnienie analizy algorytmów z różnych punktów widzenia. Algorytmy, które tutaj rozważamy, są najprostsze możliwe, tj. szeregowy, scentralizowane, statyczne i dokładne. Rozważamy tutaj takie zagadnienia, jak: poprawność, złożoność czasowa, złożoność pamięciowa i jej związki ze złożonością czasową, optymalność w sensie efektywności, stabilność numeryczna, prostota i wrażliwość. Rozdział zamykamy przykładem algorytmu probabilistycznego.

Ostatni rozdział 3 przedstawia podstawowe struktury danych, gdyż są one niezbędnym komponentem każdego rozwiązania algorytmicznego. W rozdziale tym rozważamy takie

struktury, jak: tablica, lista, zbiór, a zwłaszcza graf. Strukturom grafowym poświęcamy szczególnie wiele miejsca, gdyż grafy są najczęściej spotykanym modelem matematycznym w informatyce. Więcej informacji na ten temat można znaleźć w podręczniku Krzysztofa Goczyły „Struktury danych” [11]. Co ważne i cenne dla czytelników studiujących zagadnienia złożoności obliczeniowej algorytmów, każdy z powyższych rozdziałów kończy się zestawem ponad 30 zadań niezbędnych do sprawdzenia nabytej wiedzy i umiejętności oraz umożliwiających jej pogłębienie.

Skrypt kończymy dodatkiem, który zawiera sekwencję około 40 zagadek algorytmicznych, które wymagają znalezienia efektywnych algorytmów rozwiązujących postawione problemy. Czytelnik, przystępując do ich rozwiązywania, będzie miał okazję, aby sprawdzić swoje umiejętności i kompetencje nabyte na zajęciach z przedmiotu „Podstawy analizy algorytmów”. Wreszcie na sam koniec zamieszczamy słownik polsko-angielski ważniejszych pojęć z tego zakresu wiedzy.

W tym miejscu pragnę wyrazić wdzięczność recenzentowi prof. dr. hab. inż. Krzysztofowi Goczyły za życzliwe sugestie; dziękuję również mgr. inż. Janowi Wojtkiewiczowi za pomoc edytorską oraz zespołowi Wydawnictwa PG za wnikliwą korektę. Z góry dziękuję również studentom za wszelkie uwagi merytoryczne, które można kierować pocztą elektroniczną na podany niżej adres.

Gdańsk, kwiecień 2025 r.

Marek Kubale
kubale@eti.pg.edu.pl

SŁOWNIK POLSKO-ANGIELSKI

A

- algorytm*, 38
 - algorytm aproksymacyjny*, 105
 - algorytm Las Vegas*, 72
 - algorytm Monte Carlo*, 72
 - algorytm online*, 105
 - algorytm probabilistyczny*, 72
 - algorytm pseudowielomianowy*, 105
 - algorytm rekurencyjny*, 24
 - algorytm uczący się*, 105
 - algorytm zachłanny*, 105
 - algorytm z wyprzedzaniem*, 105
- *algorithm*
 - *approximation algorithm*
 - *Las Vegas algorithm*
 - *Monte Carlo algorithm*
 - *online algorithm*
 - *probabilistic algorithm*
 - *pseudopolynomial algorithm*
 - *recursive algorithm*
 - *learning algorithm*
 - *greedy algorithm*
 - *look-ahead algorithm*

B

- bisekcja w tablicach 2-wymiarowych*, 105
- *bisection in two-dimensional arrays*

C

- całkowita poprawność*, 40
 - częściowa poprawność*, 40
- *full correctness*
 - *partial correctness*

D

- dane istotne*, 49
 - długość programu*, 58
 - dziel i zwyciężaj*, 25
- *essential data*
 - *program length*
 - *divide-and-conquer*

E

- element*, 82
 - eliminacja Gaussa*, 55
- *item*
 - *Gaussian elimination*

F

- faktoryzacja*, 14
 - FIFO*, 87
 - funkcja iloczynowa*, 25
 - funkcja monotoniczna*, 28
 - funkcja wiodąca*, 25
- *factorization*
 - *First In First Out*
 - *multiplicative function*
 - *monotonic function*
 - *driving function*

G

- gra o sumie zerowej*, 105
 - graf niezgodności*, 49
 - graf przepływu sterowania*, 59
- *zero-sum game*
 - *incompatibility graph*
 - *program control graph*

H*harmoniczna liczba chromatyczna*, 100– *harmonious chromatic number**hipoteza Goldbacha*, 12– *Goldbach Conjecture***I***indeks stabilności (kumulacji)*, 56– *stability index**inicjalizacja wirtualna*, 85– *virtual initialization***J***jednoznacznie określona*, 38– *uniqually determined**jednostka funkcjonalności*, 59– *function point***K***kolejka*, 87– *queue**kolejka podwójna*, 87– *double queue**kompromis przestrzenno-czasowy*, 54– *trade-off between space and time**kwadratowa*, 61– *quadratic***L***leksem*, 58– *token**liczba cyklomatyczna*, 59– *cyclomatic number**liczba przecięć*, 89– *crossing number**licznik*, 89– *counter**LIFO*, 87– *Last In First Out**liniowy*, 61– *linear**lista*, 87– *list**lokalnie monotoniczny*, 93– *locally monotonic***M***macierz incydencji*, 94– *matrix of incidence**macierz rozrzedzona*, 50– *sparse matrix**macierz sąsiedztwa*, 94– *matrix of adjacency**marszruta*, 96– *route**memoizacja*, 67– *memoization**metoda sprężynowa*, 89– *spring embedder***N***niestabilny*, 56– *unstable**nietrywialna*, 96– *nontrivial**nieuchwytna*, 96– *elusive**niewielomianowa*, 62– *non-polynomial**niezmiennik pętli*, 41– *loop invariant*

O

- objętość programu*, 58
- obliczalność*, 8
- oczekiwana liczba błędów*, 58
- oczekiwana złożoność obliczeniowa*, 45
- oczekiwana złożoność pamięciowa*, 48
- operacja*, 37
- optymalizacja wielokryterialna*, 105
- optymalny*, 51
- *program volume*
- *computability*
- *expected number of errors*
- *expected complexity*
- *expected space complexity*
- *operation*
- *multicriteria (multiobjective) optimization*
- *optimal*

P

- pesymistyczna złożoność obliczeniowa*, 44
- pęki wyjściowe*, 98
- pierwszy rząd*, 27
- planarny*, 61
- planarny w górę*, 90
- podłoga (liczby)*, 17
- podstawowy*, 42
- pojedyncza liniowa (lista)*, 85
- polilogarytmiczna*, 61
- poziom programu*, 56
- półalgorytm*, 39
- problem algorytmiczny*, 9
- problem Collatza*, 12
- problem decyzyjny*, 10
- problem kafelkowania*, 11
- problem komiwojażera*, 14
- problem liczb gradowych*, 12
- problem mostów królewieckich*, 10
- problem niealgorytmiczny*, 11
- problem NP-trudny*, 103
- problem optymalizacyjny*, 10
- problem półrozstrzygalny*, 10
- problem przypuszczalnie niealgorytmiczny*, 11
- problem przypuszczalnie wykładniczy*, 14
- problem stopu*, 10
- problem wielomianowy*, 15
- problem wykładniczy*, 13
- profilowanie*, 38
- programowanie dynamiczne*, 105
- programowanie strukturalne*, 40
- przeszukiwanie liniowe z dwoma wartownikami*, 105
- *worst-case complexity*
- *forward star*
- *first-order*
- *planar*
- *upward planar*
- *floor*
- *basic*
- *linear singly connected*
- *polylogarithmic*
- *program level*
- *semi-algorithm*
- *algorithmic problem*
- *Collatz Problem*
- *decision problem*
- *tiling problem*
- *travelling salesman problem*
- *hailstone numbers problem*
- *Königsberg bridge problem*
- *nonalgorithmic problem*
- *NP-hard problem*
- *optimization problem*
- *semidecidable problem*
- *presumably nonalgorithmic problem*
- *presumably exponential problem*
- *halting problem*
- *polynomial problem*
- *exponential problem*
- *profiling*
- *dynamic programming*
- *structural programming*
- *linear search with two sentinels*

przeszukiwanie wyczerpujące, 105
przeszukiwanie wykładnicze, 105
punkt Feynmana, 13

– *exhaustive search*
– *exponential search*
– *Feynman point*

Q

quasi-liniowy, 61

– *quasilinear*

R

reprezentacja Fary’ego, 89
rozmiar, 87
rozwiązanie jednorodne, 25
rozwiązanie ogólne, 25
rozwiązanie szczegółowe, 25
równanie diofantyczne, 12
rysowanie grafów klastrowych, 94
rysowanie hierarchiczne, 89
rysowanie konfluentne, 93

– *Fáry embedding*
– *size*
– *homogeneous solution*
– *general solution*
– *particular solution*
– *Diophantine equation*
– *cluster graph drawing*
– *hierarchical drawing*
– *confluent drawing*

S

schemat Hornera, 77
skończoność, 38
skorpion, 101
sortowanie przez scalanie, 60
stała, 61
stopień trudności (programu), 58
stos, 87
struktury danych, 47
subliniowa, 61
sufit (liczby), 17
superwielomianowa, 61
superwykładnicza, 61
szybkie mnożenie macierzy, 68

– *Horner's method*
– *finiteness*
– *scorpion*
– *mergesort*
– *constant*
– *program difficulty*
– *stack*
– *data structures*
– *sublinear*
– *ceiling*
– *superpolynomial*
– *superexponential*
– *fast matrix multiplication*

Ś

śląd macierzy, 50
średni przypadek, 45

– *trace*
– *average case*

T

tablica, 84
teoria obliczeń, 10
teoria złożoności obliczeniowej, 10
token, 58

– *array*
– *computability theory*
– *computational complexity theory*
– *token*

U

ujście, 96

uruchamianie, 39

– *sink*– *debugging***W**

wektor, 84

wektor charakterystyczny, 87

wewnętrzna złożoność problemu, 52

wielkie twierdzenie Fermata, 12

wieże w Hanoi, 14

wielomianowa, 61

własność monotoniczna, 96

własność redukcji, 25

własność stopu, 40

w miejscu, 48

wołanie przez adres, 67

wołanie przez wartość, 66

wrażliwość numeryczna najgorszego
przypadku, 60wrażliwość numeryczna średniego
przypadku, 60

wskaźnik uwarunkowania, 60

wykładnicza, 61

wysiłek programisty, 58

wzory Vieta, 55

wzór Stirlinga, 32

– *vector*– *characteristic vector*– *inherent problem complexity*– *Fermat's Last Theorem*– *Towers of Hanoi*– *polynomial*– *monotonic*– *reduction property*– *halting property*– *in place*– *call-by-reference*– *call-by-value*– *worst-case sensitivity*– *average-case sensitivity*– *condition number*– *exponential*– *programmer effort*– *Vieta's formulas*– *Stirling approximation***Z**

zamiatanie warstwa po warstwie, 90

zbiór, 87

złożoność cyklomatyczna, 59

złożoność najgorszego przypadku, 45

złożoność średniego przypadku, 45

złożoność pamięciowa, 48

złożoność pamięciowa najgorszego
przypadku, 49– *layer by layer sweep*– *set*– *cyclomatic complexity*– *worst-case complexity*– *average-case complexity*– *space complexity*– *worst-case space complexity***Ż**

źle uwarunkowane, 55

– *ill-conditioned*

LITERATURA

- [1] Abran A., Robillard P.N.: Function point analysis: An empirical study of its measurement processes. *IEEE Trans. Soft. Eng.* 22 (1996), 895–909.
- [2] Aho A., Hopcroft J.E., Ullman J.D.: *Projektowanie i analiza algorytmów komputerowych*. Warszawa: PWN (1983).
- [3] Banachowski L., Diks K., Rytter W.: *Algorytmy i struktury danych*. Warszawa: WNT (1996).
- [4] Bini D., Lotti G.: Stability of fast algorithms for matrix multiplication, *Numer. Math.* 36 (1980), 63–72.
- [5] Conforti M., Cornuéjols G., Kapoor A., Vušković K.: Even hole-free graphs. Part II: recognition algorithm, *J. Graph Theory* 40 (2002), 238–266.
- [6] Coppersmith D.: Winograd S., Matrix multiplication via arithmetic progressions, *J. Symbolic Comput.* 9 (1990), 251–280.
- [7] *Encyclopedia of Algorithms* (M-Y. Kao, Ed.), Springer (2008).
- [8] Fawzi A i in.: *Discovering faster matrix multiplication algorithms with reinforcement learning*, *Nature* 610 (2022), 47–53.
- [9] Fomin F.V., Grandoni F., Kratsch D.: *Measure and conquer: A simple $O(2^{0.288n})$ independent set algorithm*. SODA (2006).
- [10] Giaro K.: *Złożoność obliczeniowa algorytmów w zadaniach*. Olsztyńska Wyższa Szkoła Informatyki i Zarządzania (2011).
- [11] Goczyła K.: *Struktury danych*. Gdańsk: Wydawnictwo Politechniki Gdańskiej (2002).
- [12] Herik H.J., Uiterwijk J.W.H.M., Rijswijk J.: Games solved: Now and in the future, *Art. Int.* 134 (2002), 277–311.
- [13] Kubale M.: *Introduction to Computational Complexity*. Gdańsk: Wydawnictwo Politechniki Gdańskiej (1994).
- [14] Kubale M.: *Introduction to Computational Complexity and Algorithmic Graph Coloring*. Gdańsk: Wydawnictwo GTN (1998).
- [15] Kubale M.: Potyczki algorytmiczne, czyli Alicja i Bogdan w różnych sytuacjach, *Pismo PG* (kolumna *Edukacja*) (2019–2020), (2022–2023).
- [16] Kubale M.: Mosty w Królewcu i hymn teorii grafów, *Pismo PG* 245 (2/2020), 50–52.
- [17] Kubale M.: *Rekordowe liczby pierwsze*, *Pismo PG* 264 (3/2023), 32.
- [18] Skeel R.: *Roundoff error and the Patriot missile*, *SIAM News* 25(1992), 11.
- [19] Strassen V.: Gaussian elimination is not optimal, *Numer. Math.* 13 (1969), 354–356.
- [20] Vassilievskia Williams V.: *Breaking the Coppersmith-Winograd barrier*, UG Berkeley (2011).
- [21] Yan S.Y.: *Teoria liczb w informatyce*. Warszawa: PWN (2006).
- [22] Wirth N.: *ALGORYTMY + STRUKTURY DANYCH = PROGRAMY*. Warszawa: WNT (1980).
- [23] <http://mathworld.wolfram.com/CollatzProblem.html>
- [24] <http://www.mersenne.org/>
- [25] <https://pl.wikipedia.org/wiki/Bogosort>

WYDAWNICTWO POLITECHNIKI GDAŃSKIEJ

Wydanie XXI zmienione i uzupełnione

Ark. wyd. 6,0, ark. druku 7,5, 1318/1288

Druk i oprawa: Volumina.pl sp. z o. o.

ul. Księcia Witolda 7-9, 71-063 Szczecin, tel. 91 812 09 08