

Spis treści

Przedmowa	13
Przykładowe programy w Pythonie	14
Co odróżnia tę książkę od innych podręczników?	15
Jak należy czytać tę książkę?	16
Co zostało opisane w tej książce?	17
Konwencje typograficzne i oznaczenia	20
ROZDZIAŁ 1. Zanim wystartujemy	23
Czym powinien się charakteryzować algorytm?	24
Jak to wcześniej bywało, czyli wyjątki z historii maszyn algorytmicznych	26
Jak to się niedawno odbyło, czyli o tym, kto wymyślił metodologię programowania	31
Proces koncepcji programów	32
Poziomy abstrakcji opisu i wybór języka	33
Maszyna Turinga	35
Modelowanie i realizacja algorytmów	37
Schematy blokowe	38
Przetwarzanie danych — operatory	40
Grupowanie fragmentów kodu w funkcje	40
Pobieranie lub wyświetlanie wartości	41
Iteracyjne wykonywanie kodu	42
Operatory logiczne	45
Poprawność algorytmów	47
Zadania	49
Rozwiązania i wskazówki do zadań	50
ROZDZIAŁ 2. Systemy obliczeniowe i podstawy kodowania	53
Systemy pozycyjne w pigułce	53
System dwójkowy	55
System szesnastkowy	56
System ósemkowy	57
Operacje arytmetyczne na liczbach dwójkowych	58
Kod BCD	60
Kodowanie liczb ze znakiem	62
Kod znak-moduł (ZM)	62
Kod U2 (system uzupełnienia dwójkowego)	63
Zmienne w pamięci komputera	64
Kodowanie znaków	65
Kodowanie obrazów	68
Mapy bitowe na przykładzie formatu BMP	69

ROZDZIAŁ 3. Rekurencja	75
Definicja rekurencji	75
Ilustracja pojęcia rekurencji	77
Jak wykonują się programy rekurencyjne?	78
Niebezpieczeństwa rekurencji	80
Ciąg Fibonacciego	81
Stack overflow!	83
Pułapek ciąg dalszy	85
Stąd do wieczności	85
Definicja poprawna, ale...	86
Typy programów rekurencyjnych	88
Myślenie rekurencyjne	90
Przykład 1. Spirala	90
Przykład 2. Kwadraty „parzyste”	94
Przeszukiwanie binarne	96
Uwagi praktyczne na temat technik rekurencyjnych	98
Zadania	99
Rozwiązania i wskazówki do zadań	101
 ROZDZIAŁ 4. Analiza złożoności algorytmów	 105
Definicje i przykłady	106
Jeszcze raz funkcja silnia	110
Wyszukiwanie wspólnego elementu w kolekcjach	111
Zerowanie fragmentu tablicy	116
Wpadamy w pułapkę	117
Różne typy złożoności obliczeniowej	118
Nowe zadanie: uprościć obliczenia!	120
Analiza programów rekurencyjnych	121
Terminologia i definicje	121
Ilustracja metody na przykładzie	123
Rozkład logarytmiczny	124
Przeszukiwanie binarne...	
tym razem bez matematyki wyższej!	126
Zamiana dziedziny równania rekurencyjnego	127
Funkcja Ackermanna, czyli coś dla smakoszy	127
Złożoność obliczeniowa to nie religia!	129
Techniki optymalizacji programów	130
Trochę praktyki: pomiary czasów wykonania	131
Zadania	132
Rozwiązania i wskazówki do zadań	133
 ROZDZIAŁ 5. Typy proste wbudowane	 135
Zmienne (nie zawsze) proste	136
Typy proste	137
Stałe symboliczne	138
Pojęcie referencji	139
Zasięg zmiennych	140

Napis niezmienny jest i basta!	141
Konwersje napisów na liczby (i odwrotnie)	142
Formatowanie wyników z użyciem notacji f''	143
Tablice (nie całkiem) klasyczne	147
Inicjalizacja tablic o stałym wymiarze	149
Tuple (czasem zwane krotkami)	150
Modyfikacja tupli	151
Zastosowania programistyczne	152
ROZDZIAŁ 6. Modelowanie abstrakcyjnych struktur danych	155
Szablon nowej struktury danych	156
Dokładamy logikę biznesową	161
Błędne użycie obiektów klasy i wyjątki	165
Przeciążanie operatorów arytmetycznych	166
Poszerzanie definicji modelu danych	168
Rekurencyjne struktury danych	171
Pułapki OOP w Pythonie	172
ROZDZIAŁ 7. Przykładowe realizacje wybranych struktur danych	173
Listy jednokierunkowe	175
„Tablicowa” implementacja list	200
Listy innych typów	206
Listy z iteratorem	210
Zbiory	213
Modelowanie kolekcji — podsumowanie	216
ROZDZIAŁ 8. Typy złożone wbudowane	217
Listy, czyli tablice dynamiczne	218
Metody dostępne dla list w Pythonie	221
Listy tworzone na podstawie wyrażeń	223
Zbiory	223
Zbiory tworzone na podstawie wyrażeń	228
Słowniki	228
Szybkie tablice NumPy	234
Instalacja	235
N-wymiarowe tablice NumPy	235
Tablice i macierze NumPy	236
Deklarowanie tablic i macierzy NumPy	237
Funkcje tablicowe NumPy	239
Zmiany układu i rozmiaru tablic NumPy	242
Wycinki w tablicach	244
ROZDZIAŁ 9. Struktury danych o dostępie ograniczonym	247
Stos	247
Zasada działania stosu	247
Realizacja programowa stosu	249
Sprawdzanie typu danych obiektu	251

Kolejki FIFO	251
Stery i kolejki priorytetowe	254
Sortowanie za pomocą sterty	262
ROZDZIAŁ 10. Drzewa i ich reprezentacje	263
Binarne drzewa poszukiwań (BST)	268
Drzewa binarne i wyrażenia arytmetyczne	273
Uniwersalna struktura słownikowa	279
Zajętość pamięci słownika	285
Drzewa „egzotyczne”	285
ROZDZIAŁ 11. Algorytmy przeszukiwania	287
Przeszukiwanie liniowe	287
Generyczne funkcje porównawcze	288
Przeszukiwanie binarne	289
Transformacja kluczowa (hashing)	291
W poszukiwaniu funkcji $H()$	293
Najbardziej znane funkcje $H()$	294
Obsługa konfliktów dostępu	297
Powrót do źródeł	297
Jeszcze raz tablice!	298
Próbkowanie liniowe	299
Podwójne kluczowanie	301
Zastosowania transformacji kluczowej	303
Klasyczne funkcje hashujące	303
Piszemy własną tablicę hashującą	304
Podsumowanie metod transformacji kluczowej	308
Przeszukiwanie danych w wybranych strukturach Pythona	309
ROZDZIAŁ 12. Algorytmy sortowania	311
Sortowanie przez wstawianie, algorytm klasy $O(N^2)$	312
Sortowanie bąbelkowe, algorytm klasy $O(N^2)$	314
Sortowanie szybkie (Quicksort) — algorytm klasy $O(N \log N)$	316
Scalanie zbiorów posortowanych	320
Sortowanie przez scalanie, algorytm klasy $O(N \log N)$	321
Sortowanie zewnętrzne	322
Sortowanie z użyciem bibliotek Pythona	326
Uwagi praktyczne	328
ROZDZIAŁ 13. Derekursywacja i optymalizacja algorytmów	329
Jak pracuje kompilator?	330
Odrobina formalizmu nie zaszkodzi!	332
Kilka przykładów derekursywacji algorytmów	333
Derekursywacja z wykorzystaniem stosu	337
Eliminacja zmiennych lokalnych	338
Metoda funkcji przeciwnych	340

Klasyczne schematy derekursywacji	342
Schemat typu while	343
Schemat typu if-else	344
Schemat z podwójnym wywołaniem rekurencyjnym	346
Podsumowanie	348
ROZDZIAŁ 14. Przeszukiwanie tekstów	349
Algorytm typu brute force	349
Nowe algorytmy poszukiwań	351
Algorytm KMP	352
Algorytm Boyera-Moore'a	356
Algorytm Rabina-Karpa	359
Kilka prostych zadań	362
Rozwiązania	362
ROZDZIAŁ 15. Zaawansowane techniki programowania	365
Programowanie typu „dziel i zwyciężaj”	366
Znajdowanie minimum i maksimum w tablicy liczb	367
Mnożenie macierzy o rozmiarze $N \times N$	370
Mnożenie liczb całkowitych	373
Inne znane algorytmy „dziel i zwyciężaj”	374
Algorytmy „żarłoczne”, czyli przekąsić coś nadszedł już czas...	374
Problem plecakowy, czyli niełatwe jest życie turysty piechura	375
Wydawanie reszty, czyli „A nie ma pan drobnych?” w praktyce	378
Programowanie dynamiczne	379
Ciąg Fibonacciego	381
Równania z wieloma zmiennymi	383
Najdłuższa wspólna podsekwencja	385
Najdłuższy wspólny podłańcuch	387
Heurystyki i inne techniki programowania	389
Uwagi bibliograficzne	392
ROZDZIAŁ 16. Algorytmy grafowe	393
Definicje i pojęcia podstawowe	395
Etykiety i wartości	396
Cykle w grafach	398
Sposoby reprezentacji grafów	401
Reprezentacja tablicowa	401
Słowniki węzłów	404
Listy kontra zbiory	405
Podstawowe operacje na grafach	406
Suma grafów	406
Kompozycja grafów	406
Graf do potęgi	407
Algorytm Warshalla	409
Algorytm Floyda-Warshalla	413
Algorytm Dijkstry	416
Algorytm Bellmana-Forda	418

Drzewo rozpinające minimalne	419
Algorytm Kruskala	419
Algorytm Prima	420
Przeszukiwanie grafów	421
Strategia „w głąb” (przeszukiwanie zstępujące)	422
Strategia „wszerz”	424
Inne strategie przeszukiwania	426
Problem właściwego doboru	426
Podsumowanie	431
ROZDZIAŁ 17. Matematyka i Python	433
Biblioteki naukowe dla Pythona	434
Klasyczne funkcje i metody matematyczne	434
Funkcje matematyczne NumPy	436
Poszukiwanie miejsc zerowych funkcji	437
Iteracyjne obliczanie wartości funkcji	438
Interpolacja funkcji metodą Lagrange’a	439
Różniczkowanie funkcji	441
Całkowanie funkcji metodą Simpsona	443
Rozwiązywanie układów równań liniowych metodą Gaussa	444
Uwagi końcowe	446
Wizualizacja danych z użyciem Matplotlib	447
Instalacja pakietu	447
Pierwszy wykres	448
Modyfikacje wyglądu wykresu	449
Wykresy statystyczne	452
ROZDZIAŁ 18. Kodowanie i kompresja danych	455
Kodowanie danych i arytmetyka dużych liczb	458
Metody prymitywne	458
Kodowanie symetryczne	461
Kodowanie asymetryczne	463
Kodowanie Base64	465
Obliczenia na bardzo dużych liczbach całkowitych	467
Reprezentacja dużych liczb całkowitych	467
Wyliczanie wartości modulo	470
Wybrane techniki łamania kodów	471
Jakość klucza szyfrującego	471
Metody łamania szyfrów	471
Techniki kompresji danych	473
Kompresja za pomocą modelowania matematycznego	475
Kompresja metodą RLE	476
Kompresja danych metodą Huffmana	477
Kodowanie LZW	483

ROZDZIAŁ 19. Czy komputery mogą myśleć...?	491
Przegląd obszarów zainteresowań sztucznej inteligencji (SI)	492
Systemy eksperckie	493
Sieci neuronowe	495
Reprezentacja problemów	497
Gry dwuosobowe i drzewa gier	500
Algorytm min-max	502
 DODATEK A. Python — lokalne środowisko pracy	 513
Testujemy poprawność instalacji Pythona	515
Instalator pip i biblioteki Pythona	518
Edytory do Pythona	519
Środowiska IDE (i dlaczego PyCharm)	521
IDLE	521
PyCharm	525
Visual Studio Community	530
Dokumentacja Pythona	531
Używanie zasobów GitHuba	533
 Literatura	 535
 Spis rysunków	 537
 Spis tabel	 540
 Skorowidz	 542